



# *MEDQUEST SYSTEM SOFTWARE*

---

## *MEDQUEST SYSTEM REFERENCE MANUAL*

---

*Production Version 7.00*

*June 30, 1999*

---

---

# TABLE OF CONTENTS

---

<b>OVERVIEW.....</b>	<b>1</b>
<i>Introduction.....</i>	<i>1</i>
<i>Conventions.....</i>	<i>2</i>
<b>QUICK START .....</b>	<b>3</b>
<i>Installation and Start Up .....</i>	<i>3</i>
Hardware .....	3
MedQuest Components.....	3
MedQuest Installation.....	4
Change the User Access Rights.....	4
<b>MODULE MANAGEMENT .....</b>	<b>5</b>
<i>Select a Module .....</i>	<i>5</i>
Select a Project File.....	5
Select a Module and Set the Module Paths .....	5
Log In.....	6
User IDs and Passwords.....	6
Use ODBC .....	6
Load a Module .....	6
Create a Module.....	8
Create an Empty Module.....	8
Create a Module from an Existing Module.....	8
Complete New Module Specification .....	8
Delete a Module .....	9
Rebuild a Module .....	9
Special Project Options .....	9
Select Project Options.....	10
Module Location Options (Data Entry Engine Only) .....	10
Module Location Options (Analysis Only) .....	10
General Options (Data Entry Engine/Analysis Only) .....	11
Data Entry Engine Options Only .....	11
Analysis Tool Option .....	12
Exit MedQuest.....	12
<b>DESIGN.....</b>	<b>13</b>
<i>Design - Selection Window .....</i>	<i>13</i>
<i>MedQuest Design Window .....</i>	<i>14</i>
File Menu .....	15
Design Functions.....	16
Add a Variable .....	16
Variable Identification.....	16
Variable Type.....	17
Single/Grid Variable Options.....	26
Popup Options .....	29
Additional Features .....	31
Add a Grid Variable.....	33
Variable Types.....	33
Mandatory.....	33

No Add Button.....	33
No Delete Button.....	34
Maximum # Rows Allowed.....	34
Medication Entry Criteria.....	34
Limiting Entry to Predefined Categories.....	34
Building a Customized Medications List.....	34
Preventing Duplicate Entry of Medications.....	35
Incident Entry Criteria.....	35
Add Variable (To Grid).....	35
Edit Variable (On Grid).....	36
Resize Variable (From Grid).....	36
Resize Grid Columns.....	36
Delete Variable (From Grid).....	37
Move Variable Left (On Grid).....	37
Move Variable Right (On Grid).....	37
Edit Variable Data Entry Rules (On Grid).....	37
Edit Variable Help (On Grid).....	37
Exit (Save Grid Variable).....	37
Cancel.....	38
Edit Variable Mode.....	38
Size/Move Variable Mode.....	38
Delete Variable Mode.....	39
Edit Variable Data Entry Rules Mode.....	40
Align Variables.....	41
Order/Indent/Unindent Variables.....	41
Move an Indented Variable.....	42
Changes to Indentation by Other Functions.....	42
Move/Copy Variables to Screen.....	42
Delete All Variables.....	43
Query/Add Variables.....	43
Edit Screen Sets/Screens.....	44
Utilities.....	47
Help.....	48
<i>Clinical Help (Design).....</i>	<i>48</i>
Create General Data Entry Guidelines.....	48
Edit Screen Clinical Help.....	48
Edit Variable Clinical Help.....	49
Compile Clinical Help Reports.....	50
<i>Data Entry Rules (Design).....</i>	<i>51</i>
Edit Rules (Rule Editor).....	51
Add/Delete/Edit a Rule.....	52
Current Rules.....	52
Add a Rule.....	52
Delete a Rule.....	52
Save a Rule.....	52
Order Rules.....	52
Apply Always.....	53
Apply Only on Current Screen.....	53
Retry Only.....	53
Edit Rule Grid.....	53
Complex Equation.....	54
Other Edit Rule Grid Functions.....	54

Select Rule Type .....	54
Edit Enable Variable Rule .....	56
Edit Disable Variable Rule .....	56
Edit Skip To Variable Rule .....	57
Edit Hot Key Save Value To Variable Rule .....	57
Edit After Entry Warning Rule.....	58
Edit Save Value to Variable Rule .....	58
Edit Exit Case Rule.....	58
Edit Post Processor Warning Rule .....	59
Edit Post Processor Mandatory Override Rule .....	59
Edit Save ? to Me (Always) Rule (Derived Variable).....	59
Edit Save ? to Me If Rule (Derived Variable).....	60
Edit Delete Derived Equation(s) for Me.....	60
Edit Do Not Load Screen/Subscreen Rule .....	60
Edit Disable Screen/Subscreen Rule.....	60
Edit on Entering Case Load Screen Set ? If.. Rule.....	61
<i>Utilities</i> .....	61
Edit Medications .....	61
View Medications .....	61
Maintain Medication Codes .....	62
Add/Edit/Delete Codes/Generics/Drugs .....	62
Edit Units Database.....	63
Add/Delete Groups of Units.....	63
Add/Delete Units to/From a Group .....	63
Importing Units .....	64
Edit Abbreviations .....	64
Edit Keywords Database.....	64
Add/Delete Keywords .....	64
Search for Existing Keywords.....	65
Add/Delete Links.....	65
Edit Variable Group .....	65
Edit Header Variables.....	66
Edit Data Entry Screen Top .....	66
Edit Variable Assistant Popup .....	67
Edit Case Status Window.....	67
Edit Opening Screen.....	67
Prepare the Module for an Opening Screen .....	67
Edit the Opening Screen .....	68
Object Properties To Be Entered For The Opening Screen.....	68
Options That Apply to All Objects.....	70
Options That Apply to Data Record Only.....	70
Rebuilding Object Tables .....	70
<i>Preferences</i> .....	71
General Options .....	71
Variable Enable/Disable Features .....	71
Data Entry Alternative Screens.....	71
Unique Case Identifier Properties .....	72
UTD Information .....	72
Colors.....	72
Miscellaneous 1 .....	72
Miscellaneous 2 .....	74
Medications .....	75

Exit Design.....	76
<b>DATA ENTRY .....</b>	<b>77</b>
<i>Data Entry - Opening Screen .....</i>	<i>77</i>
Data Entry - General Opening (Default) Screen.....	77
Data Entry - Custom Opening Screen .....	78
Browse the Objects.....	78
Search for an Object.....	79
Add an Object.....	79
Edit an Object.....	80
Delete an Object.....	80
Select a Screen Set .....	80
Data Entry General Features .....	80
<i>MedQuest Data Entry Window.....</i>	<i>84</i>
<i>MedQuest Custom Data Entry Window .....</i>	<i>84</i>
Enter Data for a Variable .....	84
Option (Pick One) .....	84
Option (Pick One or More).....	85
Option (External List) .....	85
Date .....	86
Time.....	86
Date/Time.....	86
String.....	87
Memo.....	87
Memo (ICD9).....	87
Memo (List of).....	88
Number (No Units).....	88
Number (Units) .....	88
Number Set .....	89
ICD9 Code.....	89
Diagram.....	89
Label .....	90
No UTD Option.....	90
Data Entry Assistive Aids.....	90
Medication Search Aid .....	90
Feet-Inches Calculator Aid .....	91
Enter Data in a Grid Variable .....	91
Enter Data in a Medication Grid.....	91
Add a Medication Using the Old MedQuest Medications Database .....	92
Add a Medication Using the Multum Medisource™ Lexicon Medications Database .....	93
Delete a Medication .....	93
Add Data to Related Variables.....	93
Enter Data in an Incident Grid .....	94
Add an Incident.....	94
Delete an Incident .....	94
Add Data to Related Variables.....	94
Special Features .....	95
Note Box .....	95
Pause Button.....	95
Variable Assistant Popup .....	95
Clear Value on a Variable Radio Button.....	95
Short Cut Data Entry.....	95

Printer Setup .....	96
Print Case Summary Report.....	96
<i>Clinical Help (Data Entry)</i> .....	96
View Clinical Help.....	96
<i>Data Entry Rules (Data Entry)</i> .....	96
Execute Data Entry Rules.....	96
Enable Variable Rule.....	97
Disable Variable Rule .....	97
Skip to Rule.....	97
Hot Key Rule.....	97
After Entry Warning Rule .....	98
Save Value to Variable Rule .....	98
Derive If Rule .....	98
Exit Case Rule .....	98
Post Processor Warning Rule .....	98
Post Processor Mandatory Override.....	99
View Post Processor Warnings .....	99
Disable Screen Rule.....	99
Do Not Load Screen Rule .....	99
On Entering Case Load Screen Set ? If.. Rule.....	100
<i>Exit Data Entry</i> .....	100
Post Processing After Exit.....	100
Case Status Window .....	100
<i>MedQuest Data Entry Engine</i> .....	101
<b>REPORTS .....</b>	<b>102</b>
View/Print Reports.....	102
Selecting a Report.....	102
Dictionary Variables Report .....	103
Dictionary Detailed Report .....	104
Dictionary Comparison Report .....	104
Clinical Help Comparison Report .....	105
Clinical Help (Compile) Report.....	105
Dictionary Survey Report .....	105
Case Summary Report .....	106
Data Analysis Variable Report .....	106
Data Entry Rules Report.....	106
Abstractor Summary Report.....	106
User Defined Reports .....	107
Designing Reports .....	107
Running Reports.....	108
ISO 11179 Report .....	108
<b>GLOSSARY OF TERMS.....</b>	<b>109</b>

---

# OVERVIEW

---

---

## INTRODUCTION

---

The MedQuest Clinical Data Collection Design System is a tool that provides the user with the ability to design and develop a data entry system. It enables the user to enter specifications into a database called a data dictionary and to use this data dictionary to build data entry systems (modules).

There are two primary steps involved in constructing a module: creating the data dictionary, and entering data. Once the design of a data entry system has been completed, the user can begin to enter data using the MedQuest suite of tools. In data entry mode, data can be entered for the variables that have been defined in the data dictionary. The user also has the capability to move between the design and data entry modes in MedQuest to add, delete, and change variables in the dictionary.

MedQuest can also be used to perform quality control on the data entry process. During data entry, the user can create a number of similar cases (reabstractions) by establishing a unique reabstraction identifier that is associated with the case being reviewed. This allows for easy validation of the data that have been abstracted.

Abstracted cases can then be compared using the MedQuest Quality (IQC) software utility. This tool gives the user the capability to run comparisons on cases and then select which of the answers is correct in the two compared cases, or to enter a new value if neither case is correct (adjudication). It also allows the user to explain the differences between the answers. Finally, it provides a variety of reports to summarize the quality of the data entered.

There are a variety of activities that need to be performed when implementing a complete data entry system. Some of these activities are typically performed by a network maintenance or systems manager. The MedQuest Manager (SMS) software utility is provided to assist with performing these database maintenance functions, e.g., the MedQuest Manager (SMS) utility can be used to repair a database if it becomes corrupted. Other activities supported in the MedQuest Manager (SMS) include importing data, exporting data, moving/copying data, deleting data, and maintaining the projects database.

The MedQuest Analysis Tool (MedQuest Analyzer) is a new addition to the MedQuest suite of software products. This analysis tool was created specifically to enable the user to build queries of the MedQuest data. The MedQuest Analyzer provides the capability to create an analysis, i.e., a set of logic that is organized into a single group, or query (SQL) of the data. For each project (module), several analyses may be created. These analyses enable the user to organize the queries into different logical sets. To begin these analyses, the user can add variables and/or derived variables, build indicators that can be presented in tabular (spreadsheet) or graphical formats, create VIEWS that are modifications to the indicators, export the indicators and clinical data, and create input statements for SAS or STATA. The user can then send the results to a printer or to an HTML file, which can be viewed in a browser.

The complete set of MedQuest tools now enables the user to: design a clinical data dictionary and data entry system; collect data; distribute data; and begin to analyze the collected data. MedQuest has been designed as an open architecture system and can play a variety of roles from specification construction to data entry. It can also be utilized with other third party tools to perform activities such as data reporting and analysis.

Finally, the MedQuest suite of tools provides the user with the ability to share his/her work. A data entry system (data dictionary) designed using MedQuest, can be reused/reviewed by others using MedQuest.

The reference manual provides context sensitive help for each part of the software. It is oriented to provide the user with a general, as well as a step-by-step explanation of the software. For additional information on how to use the software, tutorials are available.

---

## CONVENTIONS

---

Standard conventions have been adopted in the *MedQuest Reference Manual* to facilitate locating and identifying desired information. These conventions and their definitions include the following:

**Bolded Title Case** Windows, screen titles, program functions, variable names, and dialog boxes (e.g., the **Case Selection** screen, etc.).

**Title Case and Italicized** Text box or list box descriptions (e.g., the *Medication Selection* list box, etc.).

**Title Case** Menu options (e.g., the General Help menu option, etc.).

**<CAPITALS>** Menus, buttons, icons, and function keys (e.g., the HELP menu, the <CANCEL> button, the <NOTES> icon, the <T> key, etc.).

**Bolded** User entered data (e.g., Enter the password **MedQuest**).



Notes Symbol. This symbol indicates that the user should take notice of the information.



Advanced User's Symbol. This symbol indicates that the advanced user should take notice of the information.



---

# QUICK START

---

---

## INSTALLATION AND START UP

---

This section discusses the following topics:

- Hardware
- MedQuest Components
- MedQuest Installation

### Hardware

---

To operate the MedQuest Application, the following hardware and software are required:

- IBM or IBM-compatible personal computer with 80486/50 or higher processor
- VGA or VGA-compatible display monitor
- MS-DOS® version 5.0 or higher and Microsoft Windows version 3.1, 95, 98 or NT
- Microsoft or Microsoft-compatible mouse or pointing device
- Hard disk with a minimum of 12 megabytes of space
- Eight megabytes of RAM (sixteen are recommended)

### MedQuest Components

---

The MedQuest system includes the following components:

- **Installer (2 diskettes).** The installer is contained on two diskettes. It includes all of the software and utility files necessary to install and uninstall the different MedQuest components.
- **MedQuest Application (MedQuest).** This diskette contains the MedQuest application.
- **MedQuest Quality (IQC)/MedQuest Manager (SMS) Application.** This diskette contains the MedQuest Quality (IQC) and MedQuest Manager (SMS) utilities that work with MedQuest.
- **MedQuest Data Entry (MDE).** This diskette contains the MedQuest Data Entry application.
- **Libraries.** This diskette contains a number of additional utility files required by the MedQuest application.
- **Medications.** The medications and abbreviations databases are contained on this diskette.
- **Medications '98.** The Multum Medisource™ Lexicon Medications database is contained on this diskette.
- **ICD9.** A database with ICD-9-CM diagnosis and procedure codes is contained on this diskette.

- **MedQuest Analyzer.** This diskette contains the MedQuest Analyzer application.
- **MedQuest Analyzer Libraries.** This diskette contains a number of additional utility files required by the MedQuest Analyzer application.

## MedQuest Installation

---

The installation of the MedQuest application and other required components is explained in detail in the README.TXT file found on the installation diskette.



Because the MedQuest suite of tools has been designed to work with multiple modules, it is best to install the MedQuest Quality (IQC) and the MedQuest Manager (SMS) software in the same directory as MedQuest. This optimizes the MedQuest. Also, the MedQuest Manager (SMS) Import and Export functions are accessible from within MedQuest as well as the MedQuest Quality (IQC) Compare functions.

## Change the User Access Rights

---

The features available to a given user can be defined or changed in the **DAS.INI** file. The user access rights include:

**USER=SDPS.** If the user is set to "SDPS," MedQuest enables the user to access both the **Design** and the **Data Entry** features.

**USER=DATAENTRY.** If the user is set to "DATAENTRY," MedQuest enables the user to access only the MedQuest **Data Entry** feature.



The MedQuest Data Entry application gives the user the capability of allowing only the data entry features to be enabled.

---

# MODULE MANAGEMENT

---

---

## SELECT A MODULE

---

Before accessing a module, the user selects a module with which to work and identifies the location of the module and its components.

### Select a Project File

---

To begin using MedQuest, the user sets the path to a valid **Project Database** (FILES.MDB) by entering the path in the *Path to Project Database* (FILES.MDB) text box. The Project Database contains the locations of the data dictionary, clinical data, Medications/ICD9/Abbreviations Databases, and Employee Database of the modules in the project.

### Select a Module and Set the Module Paths

---

Select a desired module from the *List of Modules* list box or type the acronym that represents the desired module in the *Module Acronym* text box. After the module has been selected, indicate the paths to the following databases:

**Path/File Name to Dictionary Database**

Indicates the path and file name for the Module Dictionary file (e.g., CCPDICT.MDB or any valid file name). This is the file where all of the design information, clinical help, and abstraction guidelines are saved.

**Path to Datastore Database**

Indicates the path for the Module Datastore file (e.g., CCP.MDB). This is the file where all of the data that were entered are saved.

**Path to Medications/ICD9/Abbreviations (MEDS.MDB, MMEDS.MDB, ICD9.MDB and ABB.MDB) Databases**

Indicates the path for the Medications Database (MEDS.MDB/MMEDS.MDB), the ICD9 Database (ICD9.MDB) and the Abbreviations Database (ABB.MDB).

**NOTE:**

If any of these files are not used, the MedQuest Data Entry Engine will load without a warning. This also allows the user to distribute a module without having to include these files, thus saving space.

**Path to Users Database**

Indicates the path for the Module Employee database (EMPLOYEE.MDB).



MedQuest validates whether the path to each file is in the location indicated. Valid information is displayed in black; invalid information is displayed in red.

## Log In

---

Log into MedQuest by entering the user identification and password in the *User Name* and *Password* text boxes, respectively.

### *User IDs and Passwords*

---

The following log in identifications and passwords can be used for the MedQuest Application:

- User log in identification **MEDQUEST** and password **MEDQUEST**.
- User log in identifications **ID1** through **ID100** and passwords **ID1** through **ID100**, correspondingly.

The user log in identification and password can be added, modified, or deleted by using the MedQuest Manager (SMS) Application.

## Use ODBC

---

Data entry can also be accomplished using an ODBC database. Before selecting a module, use the **Export** function in Microsoft Access™ to load the datastore tables from the Microsoft Access™ database into the ODBC database. Return to MedQuest, select the module, and check the *Use ODBC* check box. Enter a path and file name in the Path to Data text box or click on the file folder at the right of the text box and select an ODBC data source available for the current configuration. It is the user's responsibility to configure the ODBC settings.

### **NOTES:**

When designing/editing a module, the ODBC option must be unselected before loading the module. If changes are being made to an existing module for which ODBC data collection has already been entered, the user will need to re-export the datastore tables into the ODBC database to keep it current.

The ODBC function in MedQuest will allow the user to enter data and report on that data. Other functions (Import, Export, Load, Archive, IQC, and Analysis) are not available when using an ODBC data source.

If ODBC has been chosen for a module, it will also be the default choice when using the Data Entry Engine software.

A data source with more than approximately 100 fields may fail in ODBC because of Microsoft ODBC constraints. If this occurs, reduce the number of fields per datastore table to enable the ODBC functionality. This is accomplished by accessing the module in MedQuest through the **Design** mode. From the FILE menu item select the Preferences option and click on the **Miscellaneous 2** tab and enter a number into the *# of Fields per Table (Max 250)* text box..

## Load a Module

---

To perform design or data entry on a given module, the user is first required to load the selected module. As the module is loading, MedQuest will verify all of the files. If there are any validation problems, a warning will be displayed.

The validation will also verify dates for Y2K compliance. If there are any date variables with formats of MM/DD/YY or YY/MM/DD, the following messages will be displayed by MedQuest:

- Module XXX (XXX being the three letter module acronym) is not Y2K compliant. Only Y2K compliant modules can be used in this (and later) versions of MedQuest. You must use an earlier version of MedQuest to change the following variable formats to Y2K compliant four digit year formats:
- Lists the variable(s) and incorrect date format (MM/DD/YY or YY/MM/DD) that must be changed to be Y2K compliant.
- If you have existing data, please make sure that you also convert that data to contain the correct four digit year corresponding to the new format you create in MedQuest. MedQuest cannot be used to correct your data and you should take steps to ensure the quality and consistency to the Y2K standard for any changes necessary.
- The program will end when you select the <CLOSE> button.

In addition, the units in CLINDATA.MDB are added to a module dictionary when a new module is created. For existing modules, a message from MedQuest will be displayed stating that the units from CLINDATA.MDB have been copied into the dictionary. After clicking on the <OK> button, all units are saved in the dictionary.



#### **Advanced User: Validation On Load**

The validation compares the data dictionary database to the datastore database. If the dictionary does not match the data that are to be stored, the user will be prompted to rebuild the system. This typically occurs when the dictionary has become corrupted or when the user has pointed to a dictionary and a datastore database that are different versions. Since the rebuild function will delete any data stored in the existing datastore database, backup the data before performing a rebuild.

#### **EXAMPLE:**

While designing Module DIA, the computer is accidentally turned off after adding variable XYZ. When the DIA module is reloaded, a warning appears telling the user to rebuild the module. This is because the dictionary file contains the variable XYZ but the datastore database DIA.MDB does not. When rebuilding, the file DIA.MDB will be rebuilt and XYZ will be added to it.



#### **Advanced User: Adding The Units From The CLINDATA.MDB**

By adding the units tables to a dictionary, any additions, deletions, or changes made to these tables will be captured in that module's dictionary. When sharing work with other users, it is no longer necessary to update the CLINDATA.MDB and send this file with the dictionary since the units are now included in the dictionary.



#### **Advanced User: Load A Module Not Listed In The Project File**

The user can load a module even though it is not included in the list of modules. This occurs frequently when another developer sends a module that is not listed in the current project file or when a module is developed under one project to be pointed to by another project. When the module is loaded, it is added to the current project file if it is a valid database.

#### **EXAMPLE:**

To load the DIA module sent from a remote site, type in DIA as a module, set the paths, and load the module. DIA will be added as a module to the current project file.



### Advanced User: Manage Projects And Modules In MedQuest/MedQuest Quality (IQC)/ MedQuest Manager (SMS)

When the Libraries diskette is installed, a new project file FILES.MDB is created. This project file contains a list of all the modules for the project. If it is necessary to work with multiple projects, do so by using multiple copies of the file FILES.MDB.

MedQuest is currently the only utility that can be used to create modules, add modules to the project database, and delete modules from the project database. In both MedQuest and MedQuest Manager (SMS), the user can change the paths that point to the components of the module. In MedQuest Quality (IQC), the user can only view the modules and paths listed in the project database. It is important to have a strategy for organizing projects and modules and their component parts before working in either MedQuest design or data entry in a multi-user environment.

## Create a Module

---

The user can create a new module as an empty module (no screen sets or screens) or create a new module using an existing module as a template. Both methods build a new dictionary database, a new datastore database, and add the module to the project database (FILES.MDB). From the **Log in/Manage Modules** screen, click on the <CREATE MODULE> button to display the *Create New Module* popup box and follow the instructions below.

### Create an Empty Module

---

When an empty module is created, a given set of system variables will automatically be added to the dictionary. These system variables are required by all modules to perform functions such as uniquely identifying a case, saving the dictionary version, and saving the user name.

### Create a Module from an Existing Module

---

The user can create a module from an existing module that is listed in the current project database. When this option is chosen, a new module is created that is an exact clone of the selected module. All of the screen sets, screens, and variables will be copied into the new module; only the name of the module will be changed.

### Complete New Module Specification

---

Whether creating an empty module or a module from an existing module, complete the following:

**NEW MODULE NAME.** The new module name (acronym) must be three characters (A-Z, 0-9). This allows easy tracking of a module throughout the system.

**USE 32-BIT DATABASE.** This will create a module that can be used in a system that uses 32-bit applications. Once a module has been created as a 32-bit database, it cannot be used in the 16-bit version of MedQuest. However, a 16-bit database can be utilized in the 32-bit version of MedQuest as well as the 16-bit version.

**NEW MODULE DESCRIPTION.** This can be any description.

**NEW DICTIONARY PATH/FILENAME.** A valid path and filename for the new module's dictionary must be entered. This is where the module dictionary and datastore databases will be created. The user can enter any name for the dictionary file. However, it is recommended that a common naming convention such as the module name plus "DICT.MDB" be used.

**NAME THE MODULE DICTIONARY FILE.** The user has the capability to select any name for the dictionary file. This has been designed for future flexibility to allow multiple modules to reside in one physical dictionary file and to eventually allow a master dictionary to be accessed by multiple users across a wide area network. However, it is recommended that the user continue to uniquely identify each dictionary file using the module name plus "DICT.MDB."

After all of the required information has been completed, click on the <OK> button to save the new module or the <CANCEL> button to cancel the module.

## Delete a Module

---

When a module is deleted, MedQuest deletes the module name entered on the screen. This will delete the module from the project database (FILES.MDB) but it **will not delete** any of the module's files (data dictionary and datastore databases).

## Rebuild a Module

---

Whenever MedQuest warns the user to rebuild a module, it is because the data dictionary and datastore databases are no longer in the same version. This function is required before a module can be loaded. It is important to backup the datastore database or rely on the MedQuest backup.

### EXAMPLE:

When the DIA module is rebuilt, MedQuest will create a new datastore database DIA.MDB with no data, and will backup the old DIA.MDB to DIA.BAK.



### Advanced User: How the Rebuild Works

The rebuild reads the data dictionary field by field. It builds a table with the name of the module in the datastore database for all of the single variables, up to 250, and continues to add a new table for each new set of 250 unique variables using the name of the module plus "S" and a unique integer identifier. For each multiple variable, it creates a new table including that variable (and any variable related to it) using the name of the module plus "ML" and a unique integer identifier.

### EXAMPLE:

If the DIA module has 320 variables, including 2 multiples with 4 related variables each, the rebuild function will create DIA.MDB with a DIA table (250 single variables), DIAS1 table (60 additional single variables), a DIAML1 table (5 variables for the first multiple including its 4 related variables) and a DIAML2 table (5 variables for the second multiple including its 4 related variables).

## Special Project Options

---

When the user is distributing a module to another site and only wants to send them the data entry engine (no design allowed), he/she can take advantage of one or more of the following features.

## Select Project Options

---

Once a valid **Project Database** (FILES.MDB) has been selected, edit the Project Options by clicking on the <PROJECT OPTIONS> button on the **Log In/Manage Module** screen. This will display the **Edit Project Options** screen.

When distributing an application to another user, the user can distribute the Data Entry Engine and/or the Analysis Tool with certain restrictions. There are several options available for each of these applications.

### NOTE:

These options are saved in the Project Database: C:\MDQ400\FILES.MDB (designated by the user as the directory). Distribute this FILES.MDB (along with the dictionary and datastore databases) to the other designated users along with the MedQuest Installer, Data Entry Engine, Libraries, and, if applicable, the Analysis, Analysis Libraries, Medications, and ICD9 disks.

### NOTE:

After making changes, it will be necessary to restart the application to update the new FILES.MDB.

## Module Location Options (Data Entry Engine Only)

---

**DEFAULT (USE MODULE PATHS IN THE FILES.MDB, DO NOT ALLOW CHANGES TO MODULE PATHS).** The user only has access to the files listed in the FILES.MDB. The user cannot change any of the database paths.

**USE MODULE PATHS IN THE FILES.MDB, ALLOW CHANGES TO MODULE PATHS.** The user has the capability to change the database paths.

**SEARCH LOCAL DIRECTORY FOR EXISTING MODULES (DO NOT LIST MODULES IN FILES.MDB).** Only the modules located in the local directory will be listed. The user will only have access to this directory and will not be able to change the database paths.

### NOTE:

This assumes that the dictionary, datastore, medications, and employee databases are all in the directory where the MQDATA.EXE resides.

## Module Location Options (Analysis Only)

---

**DEFAULT (USE MODULE PATHS IN THE FILES.MDB, DO NOT ALLOW CHANGES TO MODULE PATHS).** The user only has access to the files listed in the FILES.MDB. The user cannot change any of the database paths.

**USE MODULE PATHS IN THE FILES.MDB, ALLOW CHANGES TO MODULE PATHS.** The user has the capability to change the database paths. Choose this option if the user has only been given **Analyze** capabilities.

**SEARCH LOCAL DIRECTORY FOR EXISTING MODULES (DO NOT LIST MODULES IN FILES.MDB).** The FILES.MDB will list all of the modules located in your local directory. The user will only have access to this directory and will not be able to change the database paths.



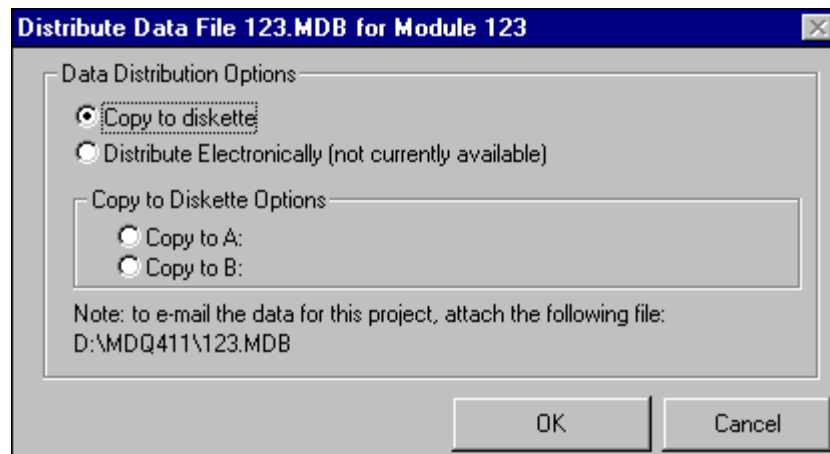
**NOTE:**

This assumes that the dictionary, datastore, meds, and employee databases are all in the directory where the ANALYSIS.EXE resides.

## General Options (Data Entry Engine/Analysis Only)

---

**DISPLAY DATA DISTRIBUTION BUTTON.** This option will enable the <DISTRIBUTE DATA> button on the **Log In/Manage Module** screen. Clicking on the <DISTRIBUTE DATA> button will display the *Distribution Data File* popup box.



Currently, the only available option is Copy to Diskette. The Distribute Electronically option will be available in a future release. By default, the Copy to Diskette option is enabled. Select the directory location from the Copy to Diskette Options and click on the <OK> button to copy the datastore database to diskette.

**NOTE:**

To e-mail the data for a specific project (i.e., CCP), attach the datastore dictionary file CCP.MDB.

**NO LOGIN SCREEN.** This option will not display the **Log In/Manage Modules** screen. The user will be taken directly into the data entry system. This is very advantageous if the developer is sending out a single data abstraction tool. The user will not have access to any other files and will perform data entry on the distributed module only.

**DO NOT DISPLAY MODULE PATHS.** This option will display the **Log In/Manage Modules** screen without displaying the paths to the database files.

This feature can only be utilized for the Analyzer application when the Allow Analysis Only (No Design) option has also been selected. For the Data Entry Engine application, either the Default option or the Search Local Directory option must be selected.

## Data Entry Engine Options Only

---

**NO REABSTRACTIONS.** This option will prevent the user from creating reabstraction cases during the data entry process. When there is no need to abstract data for a case more than once for quality control purposes, e.g., to compare the answers entered by two different data abstractors, select this option. This option is useful when the collected data are for informational purposes only.

**ALLOW DATA RECORDS TO BE DELETED.** This option will allow the user to delete records in the data entry engine.

**ON LOAD, AUTOMATICALLY REBUILD (WITH DATA COPY) FOR NEW DICTIONARY VERSION.** This option will automatically rebuild the dictionary and copy the data in the datastore database into the new dictionary. This option is necessary when the Display Only Project List/Description is selected and a new version of a module is being distributed. The user will not have the capability to rebuild the module since the <REBUILD> button is not displayed. Selecting this option ensures that the data that has already been abstracted is copied into the new datastore database.

**ONLY ADD/EDIT FOR DATA RECORDS.** This option will prevent the user from adding objects or editing data objects (other than data records) when a custom Opening screen is used.

**DISPLAY IQC FUNCTION MENU.** This option will allow the user to access the MedQuest Quality functions from the data entry engine.

## **Analysis Tool Option**

---

**ANALYSIS TOOL: ALLOW ANALYSIS ONLY (NO DESIGN).** This option will provide the user with access to the Analyze feature only.

## **Exit MedQuest**

---

Exit MedQuest by selecting the Exit option from the FILE menu item in design or data entry, and then exit the MedQuest system by clicking on an <EXIT> button or by selecting the Exit option from the FILE menu item.

---

# DESIGN

---

---

## DESIGN - SELECTION WINDOW

---

After a module has been loaded, select the Design option on the next window that appears. After selecting the MedQuest Design option, there are a number of functions that can be performed. These functions are all activated from icons or drop-down menus available on the screen:

**DESIGN A DICTIONARY.** Design a dictionary by selecting the <OK> button. If there is more than one screen set in the module, select the screen set to design and then continue.

**SELECT MODULE.** To select another module to design, click on this button.

**VIEW/PRINT REPORTS.** Produce a number of reports from the dictionary: a list of variables; a detailed dictionary report; a comparison report that produces a list of discrepancies between two different versions of a given module; a survey that can be used to collect the data for a module on paper; a data analysis by variable report; a report listing the data entry rules; and a clinical help report. This report function also allows the user to compile a Microsoft help file (.HLP) that organizes the help by screen set, screen, and variable. In addition to reports from the dictionary, this function enables the user to produce a case summary report of the data that have been entered during data entry as well as an abstractor summary report. Additional reports provide the user with the capability to design reports based on his/her specific needs and ISO reports that display all the variables that had ISO attributes assigned during the design process.

### EXAMPLE:

Provide a paper data entry form to mail to 10 data abstractors at a hospital site. Select the Dictionary Survey report after building a module. This produces a paper data collection instrument that matches the data screens to be completed using the MedQuest **Data Entry** function.

**QUERY VARIABLES.** This function enables the user to browse through the variables and their clinical help for all of the modules listed in a project database. There are three search mechanisms available: the user can search for a variable by module when the screen and variable are known; through all of the variables in a selected module by using a word or phrase that might be included in the variable's title; and using the keywords that have been associated with a variable in a given module's dictionary. These keywords can be added during module design to create a more robust dictionary variable classification scheme.

### EXAMPLE:

Someone calls and inquires about the clinical help for heart-related variables in a dictionary. While on the phone, the user can open the module and use the <QUERY> button to locate those variables and browse the help.

**IQC.** Access the **MedQuest Quality** functions by clicking on the IQC menu item and selecting one of the following options: Single Compare and Load (Original/Reab); Load (Original/Reab); Single Compare (Original/Reab); and Select Option/Comparison Set. These functions are briefly discussed in the Data Entry section, while complete guidelines are available in the MedQuest Quality Reference Manual.

**HELP.** Access the General Help by clicking on this button. The General Help provides general MedQuest application help. Other available help includes the Module Help, General Data Entry

Guidelines, and About (MedQuest) from the HELP menu. If help has been compiled for a dictionary, that help file (XXXDICT.HLP where XXX is the module acronym) can be viewed through Module help. The General Data Entry Guidelines help is a placeholder for the module developer to create this help file. (See the Clinical Help section for details on how to create this help file.) About (MedQuest) identifies software and dictionary version information.

**EXIT.** Exit MedQuest by selecting the <EXIT> button.

---

## MEDQUEST DESIGN WINDOW

---

When starting the design process, the **Design** window is displayed. The components are as follows:

**TITLE BAR.** Displays the MedQuest function, **Design**, and the module acronym on which the user is working.

**MENU BAR.** Displays the list of functions that can be performed on the **Design** window. Each of these functions contains additional options. Use the mouse to display the available options.

**DESIGN BUTTON BAR.** Displays the different functions that can be used to design a module. The buttons on the bar perform the same functions as the options listed under the DESIGN menu. As the user moves his/her mouse across the button bar, a small box describing each function will appear.

There are two types of **Design Action** buttons: **Direct Action** and **Action Mode**. The **Direct Action** buttons include:

- Add Variables
- Add Grid Variables
- Align Variables
- Order/Indent/Unindent Variables
- Move/Copy Variables To Screen
- Delete All Variables Mode
- Edit Screen Sets/Screens

The **Direct Action** buttons enable the user to perform the function that is described on the button when it is selected (clicked).

The **Action Mode** buttons include:

- Edit Variable Mode
- Size/Move Variable Mode
- Delete Variable Mode
- Edit Rules Mode

The **Action Mode** buttons turn on the selected action mode; they do not directly perform the function. This type of button allows the user to continue performing the selected function without having to reselect

the function button when switching to another variable. After selecting an **Action Mode** button, select a variable in the **Variable Design** window area on which to perform the function.

**TABS/SUBTABS.** Displays the names of the screens for the selected module or the selected set of screens. Each module can have up to 10 tabs, and each tab can have up to five subtabs associated with it. When selecting a tab or subtab, the **Variable Design** window opens directly below the selected tab or subtab.

Several enhancements have been added to the tabs and subtabs. When in **Design** or **Data Entry** mode, the title of the tab/subtab is bolded. A tab that displays an arrow at the end of its title indicates that there are subtabs associated with it. Other available options include designating if a screen has been partially or completely entered. This option is discussed in the Preferences section.

**VARIABLE DESIGN WINDOW.** Displays the variables, if any, that are associated with the selected screen or subscreen. On this screen, the user can perform a variety of variable design activities, depending on the current working mode. When using the **Edit**, **Size/Move**, **Delete**, or **Edit Rules** modes, clicking the left mouse button anywhere on the variable's frame will select the variable for the given mode's action. When selecting a variable's frame using the right mouse button, a popup menu will appear that lets the user edit or view the selected variable's help.

## File Menu

---

The FILE menu provides the following options:

**SAVE AS.** While designing a data entry system, the user may find it necessary to save the current version at a point in time. Selecting the **Save As** function in MedQuest will save this current version as a new module. To identify the new module, enter a new module acronym (three characters), a new module description, and the name of the file in which to save the dictionary.

Once this step has been completed, the new module will appear in the project file's list of modules where it can be accessed. The **Save As** function creates new module dictionary and datastore databases.



### Advanced User: Advantages Of Save As

While making changes to a dictionary, the user can lose track of the history of the changes. By taking advantage of the **Save As** function and the dictionary comparison report, a set of reports can be produced that enables the user to track the history of the changes.

Only one XXXDICT.BAK (XXX = module acronym) is saved at any given time. This file is saved in the same directory as the current dictionary. It is overwritten each time MedQuest is accessed or exited. To save dictionaries for historical reasons save them in a different directory.

#### EXAMPLE:

In the DIA module, a physician is redesigning the questions that appear on the **Laboratory** screen. Before entering the specifications, save the DIA module to another directory. After making the changes, produce a dictionary comparison report between the current DIA dictionary and the old DIA dictionary that was saved in another directory. Send the report to the physician to verify the entry of the requested changes.

**PREFERENCES.** The options available under Preferences are discussed in the Preferences section.

**PRINTER SETUP.** This option lets the user setup his/her printer options, i.e., select a printer and define the document properties (e.g., paper size, etc.). This option is useful when printing reports.

There are several steps to follow when capturing a screen in design or data entry for printing. To copy the screen, click on the <PRINT SCRIN> button or click on the <ALT> + <PRINT SCRIN> buttons to capture only the active window. After opening a word processing package such as Word or a graphics package such as Paint, paste the screen into the document by clicking on the <CTRL> + <V> buttons. The screen can also be copied into the document by using the paste function available in the application being used.

**EXIT (MEDQUEST).** After completing the design, exit the design session by clicking on the <EXIT> button or by selecting the Exit option from the FILE menu item. If any changes were made to the dictionary that would change the module's datastore database (where data are saved during data entry), MedQuest will rebuild the datastore database. The current datastore database will be saved to a backup file (XXX.BAK, where XXX = module acronym) before the new database is created.

If certain changes were made, the user cannot exit the design without the rebuild occurring. This is to ensure that the dictionary database is properly synchronized with the datastore database.



### **Advanced User: When Rebuild Is Necessary**

Rebuild is necessary whenever a change alters the datastore database. Listed below are the key changes that will invoke the rebuild:

- Adding/deleting/moving a variable
- Increasing the required field length for a variable
- Changing a variable's name
- Changing a variable's edit type

#### **EXAMPLE:**

Increasing the number of digits for a variable type number from 4 to 7, will force a rebuild so the datastore database has enough room to save the data entered.

## **Design Functions**

---

### **Add a Variable**

---

A single variable is the variable that is most common in the design of a data entry system. There are a number of different types of variables that are described below. All of the required information for a given variable type must be completed before MedQuest will save those changes.

### **Variable Identification**

---

**NAME.** Enter a unique name (up to 32 characters or less) to identify each variable. The user has the capability to reuse a variable (duplicate variable) on different data entry screens. By typing in the name of a variable that has already been used, a message will be displayed asking whether the user wants to retrieve the duplicate variable and its properties.

#### **NOTE:**

The capability of giving a variable a unique name using up to 32 characters is available for new modules created in MedQuest 4.10 and later. Existing modules and modules created from these existing modules cannot have their variable names edited to contain more than eight characters.

The user can also use the **Quick Copy** feature to duplicate an existing variable in a current module. This feature requires the user to select a screen set, screen, and the variable to quick copy. To copy a variable from another module, use the **Query/Copy** function.

**NOTE:**

Be careful if a variable is reused but its properties are not copied into the **Design** window. When the changes are saved, any changes made will also be saved for that variable name wherever it is used. However, if a variable is copied and it is renamed, then any changes made to that variable will not affect the variable from which it was copied.



### **Advanced User: Naming Strategies**

It is recommended that each new variable name begin with the three-letter module acronym. This is also the default when adding a variable to a module. The **Add/Edit Variable** screen will display the first three letters of the module acronym in the Name text box. This convention allows for easier manipulation of the variable during other phases of the project, particularly the data analysis phase.

**NOTE:**

Due to the restrictions in the software, the first character of a variable's name must begin with a letter. Therefore, to avoid having to erase or change the first character of every variable's name that is added to a module when it begins with a number, an option is available in Preferences that turns off this default feature. See the Preferences section for complete details.

**SHORT TITLE.** The short title (30 characters or less) is a short description of the variable that is used primarily in reports and places where space is limited. For example, the short title is used in the **Case Summary** report and is used by the MedQuest Quality (IQC) program to display the differences between two cases entered for a module. It is advantageous to have a meaningful short title.

**SCREEN TITLE.** The long title (254 characters or less) is what appears on the **Data Entry** screen. It is recommended that the user make this as concise as possible to leave enough space on the screen for other variables.

## **Variable Type**

---

Following are the types of variables that are available and the properties that are required for each type:

**OPTION (PICK ONE).** This variable type enables the user to define a list of options from which the data abstractor can select only one. Each option (254 characters or less) can be added, deleted, edited, or moved in the list. The system default for variable types Option (Pick One) and Option (Pick one or more) will display 15 or fewer options from which to choose; otherwise the option(s) can be selected from a list box. Twenty-five or fewer options per variable are recommended. It is also recommended that the option descriptions be brief to minimize the impact to the available space on the screen.

**NOTE:**

The developer has the capability of listing up to 99 options before the variable is turned into a list box. See the Preferences section for a complete description of the "Switch to a list box when number of items in Option (Pick One) or Option (Pick one or more) is greater than ?" option.

A set of default options (i.e., Yes/No, True/False, None of Above, UTD) is also provided as commonly used options. By default, the options are assigned a value to be saved (e.g., first option = 1, second option = 2, etc.) in data entry. The user has the capability to change that value. Each value that is saved can only be one or two characters in length. Note that the default options have default values that are saved for that option (Yes = 1, No = 2, True = 1, False = 2, None of Above = 0, and UTD = 0).

#### EXAMPLE:

Use Option (Pick One) to collect information on whether the patient is Male or Female. Assign "1" as a value to indicate male, "2" to indicate female, and "0" to indicate UTD.

#### NOTE:

If any option has the same value to save as another option, the user will receive a warning that all of the values to save must be unique.

The user also has the capability to disable an option or make it invisible. To disable an option, select the option from the *Options (pick one)* list box and check the disable feature. The data abstractor will see the option while in data entry but he/she will not be able to select this option. To make an option invisible, select the option from the *Options (pick one)* list box and check the invisible feature. The data abstractor will not be able to see the option while in data entry.

**OPTION (PICK ONE OR MORE).** This variable type enables the user to define a list of options from which the data abstractor can select one or more items. Each option (254 characters or less) can be added, deleted, edited, or moved in the list. The system default for variable types Option (Pick One) and Option (Pick one or more) will display 15 or fewer options from which to choose; otherwise the option(s) can be selected from a list box. Twenty-five or fewer options per variable are recommended. It is also recommended that the option descriptions be brief to minimize the impact of the available space on the screen.

#### NOTE:

The developer has the capability of listing up to 99 options before the variable is turned into a list box. See the Preferences section for a complete description of the "Switch to a list box when number of items in Option (Pick One) or Option (Pick one or more) is greater than ?" option.



The user also has the capability to disable an option or make it invisible. To disable an option, select the option from the *Options (pick one or more)* list box and check the disable feature. The data abstractor will see the option while in data entry but he/she will not be able to select this option. To make an option invisible, select the option from the *Options (pick one or more)* list box and check the invisible feature. The data abstractor will not be able to see the option while in data entry.

An additional feature, "Clear others if true," allows the user to choose the option to clear all the other options. To select the option to clear all others, highlight the item and check the *Clear Others if True* check box.

#### NOTE:

By default, if the None of Above option is selected, it will clear all the other items.



#### Advanced User: How Data Are Saved

In data entry, each option type is saved as a 0 (not marked) or a 1 (marked) in a string. The first element in this string is always associated with the last option to allow easy analysis of the option that clears all of the others. The first option is saved in the second position in the string, etc.

#### EXAMPLE:

The available options include Option 1, Option 2, Option 3, and None of Above. If option None of Above is selected, the other options will be cleared and the string will be saved as "1000." Selecting Option 1 and Option 3 will save the string as "0101." Selecting Option 2 will save the string as "0010." Selecting all of the options (except None of Above) will save the string as "0111."

**OPTION (EXTERNAL LIST).** Select this variable type when using a list of options that exceeds the MedQuest limitation for Option (Pick one) or Option (Pick one or more). This requires the user to create a Microsoft Access database containing the list of options and the values to be saved. When selecting this option, provide an external database name, a table in the database containing the available options, the name of the field that contains the options to display on the screen, and the name of the field that contains the value to save. Also required is the maximum number of selections the user is allowed to make.

Another feature, Field to Filter On (variable must also be on data entry screen), allows the user to filter on the external database options based on the response to another variable. For instance, to select only the counties in a specific state, the user could filter on a state variable to display only those related counties during data entry.

After creating the external database for a county lookup table, one of the field names should be identified as State (linking county/state). Create a variable with the same field name of State to link to the Option (External List) variable. A string variable named State specified as a State Code will insure that a valid state code is entered during data entry. Edit the Option (External List) variable and click on the drop-

down menu list for the Field to Filter On option and select the State field. These two variables are now linked by state codes. In data entry, when a state code is specified in the State variable, only the counties associated with that state would be displayed in the County (Option (External List)) variable. Note that the County variable will remain blank (or display UTD if that option is available) until the State variable has been entered.

When designing the database table, the following specifications are required:

- Field to Save field length =< 32 characters
- Field to Save field must have a unique value for each record
- Index with same name as Field to Save field must exist on the field
- Index with same name as Field to Display field must exist on the field
- Variable with Field Name to Filter By must also exist on the same data entry screen as the external list variable

#### EXAMPLE:

Use this variable type for a list of microorganisms in a database from which the data abstractor can choose.

#### NOTE:

By default, "UTD" is included as an option. If the user chooses to include "UTD" as an option, it will appear as the first item on the list, and selecting "UTD" will clear all the other items on the list.



#### Advanced User: Design Strategy

When the data are saved, each option the data abstractor selects saves the **Field to Save** value; each of these values is separated from the others by a semicolon. For example, if three options being saved have values of 001, 024 and 099, the saved string would be "001;024;099;." Since the maximum length of the **Field to Save** is 254 characters, it is important to strategically design an external database when allowing the user to select more than one option. The system will calculate the maximum number of options by multiplying the number of allowed options by the length of the **Field to Save** plus 1. If this result exceeds 254, a warning is given that too many options are being allowed.

It is recommended that the **Field to Save** field length be 2 or 3 characters. Another good design practice is to keep the number of options in the database to 400 or less (limit is 6000 options). The current design loads all of the options in the list; therefore, the longer the list, the more time it takes to load.

It is also recommended that the external database table be created inside the module dictionary (XXXDICT.MDB) to simplify the copying/transferring of dictionaries.

**DATE.** This variable type requires that the user select the format in which the date will be entered. There is a list of allowed formats from which to select. These formats apply only to data entry. When the data are saved, all dates without a century requirement are saved as MM/DD/YYYY. All dates with a century requirement are saved as MM/DD/YYYY. The default format for a date variable is MM/DD/YYYY.



**Date, Date/Time Options**

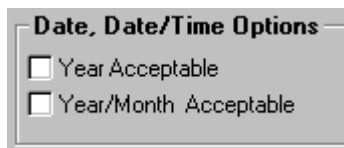
☐ Year Acceptable

☐ Year/Month Acceptable

An additional option available for the Date and Date/Time variables is the capability to save a date using only the year or the month and the year. When creating this type of variable, select either the Year Acceptable option or the Year/Month Acceptable option. In data entry, the data abstractor would be able to save a partial date instead of marking the date as "UTD."

**TIME.** This variable type requires that the user select the format in which the time will be entered. There is a list of allowed formats from which to select. These formats apply only to data entry. When the data are saved, times are saved as HH:NN if no seconds are included or as HH:NN:SS if seconds are included in the format. Data are saved using military time. For example, if a time variable is entered as 3:15 PM, that data will be saved as 15:15. If PM has not been specified during design, and the time is entered as 3:15, the time will reflect 3:15 AM.

**DATE/TIME.** This variable type requires that the user select the format in which both the date and time will be entered. There is a list of allowed formats from which to select. These formats apply only to data entry. When the data are saved, if the format requires a century and seconds, the data will be saved as MM/DD/YYYY HH:NN:SS, otherwise it will be saved as MM/DD/YYYY HH:NN. The default format for a date/time variable is MM/DD/YYYY and HH:NN.



**Date, Date/Time Options**

☐ Year Acceptable

☐ Year/Month Acceptable

An additional option available for the Date and Date/Time variables is the capability to save a date using only the year or the month and the year. When creating this type of variable, select either the Year Acceptable option or the Year/Month Acceptable option. In data entry, the data abstractor would be able to save a partial date instead of marking the date as "UTD."

**STRING.** This variable type enables the data abstractor to enter any string of characters. The designer is required to enter the length of the string up to 99 characters. A feature "Require all digits" can be selected when creating this type of variable to force entry of all of the digits.

**EXAMPLE:**

Social security numbers are nine digits long. If this option has been selected, the data abstractor must enter all nine digits.

**String Options**

☒ Upper Case    ☐ State Code (must be 2 digits)

**Custom Edit Mask**

(###) ###-####

Defined characters you can enter into mask above

- # = allow number only
- & = allow any character
- ? = allow letters only
- A = allow numbers and letters
- U = allow letters only and make uppercase
- L = allow letters only and make lowercase
- 0-9 = user defined below

0=	
1=	
2=	
3=	

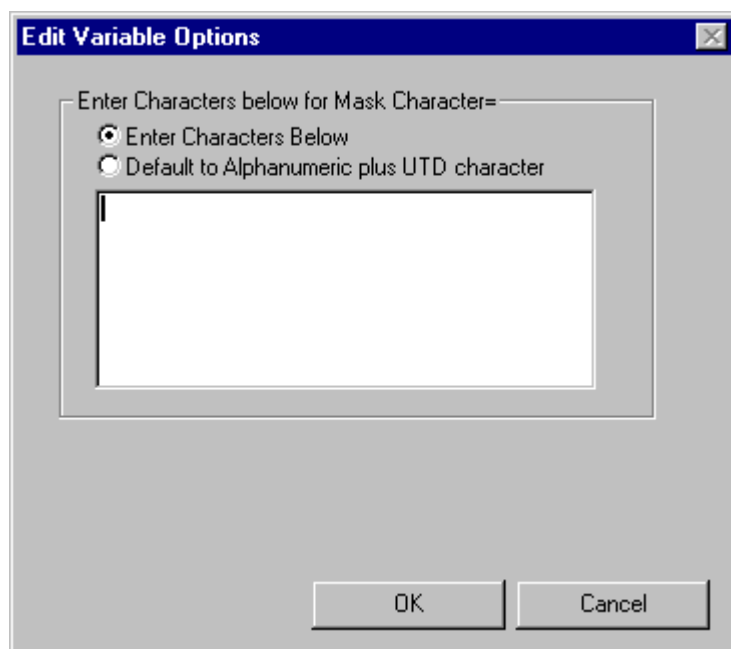
Any other character will not allow user to type over  
Example: (###)###-#### for telephone number

Additional features include: **Upper Case** and **State Code (Must be 2 Digits)**. When **Upper Case** is selected, the data entered will be displayed in upper case only. Note that the Upper Case option will also work in conjunction with the custom edit mask. For example, if the "& = allow any character" Edit Mask option and the Upper Case option have been selected, all letters entered in data entry would be displayed and saved in upper case.

Selecting the **State Code (Must be 2 Digits)** feature will ensure the entry of a state code as two uppercase digits (i.e., VA [Virginia]) as well as a valid state code. When this option is selected, the designer will be required by MedQuest to enter "2" in the *Length of String* text box, check the Require All Digits option, and select the Upper Case option in the *String Options* text box.

The designer may add a custom edit mask for formatting data such as telephone or social security numbers. The existing edit masks are: # - allow number only; & = allow any character; ? = allow letters only; A = allow numbers and letters; U = allow letters only and make uppercase; L = allow letters only and make lowercase; 0-9 = user defined. Any other character will not allow the user to type over it. For example, to create a telephone number, enter (###) ###-#### in the *Custom Edit Mask* text box. Note, the abstractor can only enter as many characters as were defined in the mask in the order in which they were defined.

To specify a user-defined character, highlight a number (0-9) in the *User-defined* text box, and type the character(s) in the popup box. Choosing the default radio button will allow alphanumeric plus the UTD character.



When designing a mask and using the user defined character in the custom edit mask string, the data abstractor can only enter the defined characters in the order in which they were defined.

#### EXAMPLE:

For a diagnosis code, the user could define the character "0" as "V, v, E, e, or a blank space that will allow a V, an E, or a space to be entered. The designer can then use this defined character in the custom edit mask as 0###.## (i.e., V123.12, e123.12, or 123.12).

**MEMO.** This variable type allows the user to enter up to 32,000 alphanumeric characters. Since the memo data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

#### NOTE:

Since this is a free form text field, the data collected will not be conducive for analysis.

**MEMO (ICD9).** This variable type allows the data abstractor to enter up to 32,000 characters. In addition, the ability to retrieve and embed an ICD-9-CM diagnosis code (and description) or an ICD-9-CM procedure code (and description) is provided. Select the type of code to provide from the **Edit Variable** window.

The user also has the option of saving the decimal point associated with each ICD9 code. To select this feature, click on the Include *Decimal Point for Code* check box located beneath the choices for the ICD9 variable. A code originally saved as 30089 in the datastore database will now be saved as 300.89. If during the design process the option to include a decimal point associated with an ICD9 code is selected, it will be displayed on the **Data Entry** screen after a selection has been made.

During data entry, the data abstractor will be provided with an assistive device that allows him/her to retrieve the codes. Since the memo data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

#### NOTE:

Since this is a free form text field, the data collected will not be conducive for analysis.

**MEMO (LIST OF).** This variable type allows the data abstractor to enter data for each item in a list of memos. The user can add one or more memo titles for the data in the list of memos provided. The abstractor can enter a total of 32,000 characters for all of the items in the list combined. Since the memo data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

**NOTE:**

Since this is a free form text field, the data collected will not be conducive for analysis.

**NUMBER (NO UNITS).** The number with no units requires the user to specify the maximum number of whole digits and decimal place digits that the data abstractor is allowed to enter. Leaving the *# Decimal Digits* text box blank will allow the data abstractor to place the decimal anywhere within the number. Entering 0 (zero) in the *# Decimal Digits* text box will prevent the data abstractor from entering a decimal place. During entry of a number, the data abstractor will be allowed to enter only one decimal place at any location within the number.

A feature "Require all digits" can be selected when creating this type of variable to force entry of all of the digits.

An additional feature, "Optional Allowed Code Descriptions," lets the designer define each option and assign to it a value to be used for data entry and as the value to save. The numbering of these codes must be sequential, starting with 0 or 1 and continuing up to 99. This feature will require the data abstractor to enter only the defined codes for each option.



**Advanced User: Optional Allowed Code Description**

If there are several variables that require the same optional answers, it is advantageous to list the directions and required responses only once. In data entry, only the allowable answers will be accepted.

Test for Balance	
(Code for ability during the last 7 days)	
0. Maintained position as required in test.	
1. Unsteady, but able to rebalance self without physical support.	
2. Partial physical support during test; or stands (sits) but does not follow directions for test.	
3. Not able to attempt test without physical help.	
a. Balance while standing	a. Balance while sitting
<input type="text"/>	<input type="text"/>

In the example above, the data abstractor could only enter 0, 1, 2, or 3 for variables A and B. This feature is also very useful for data analysis purposes.

**NUMBER (UNITS).** For a number with units, the user must specify the maximum number of whole digits and decimal place digits that the data abstractor is allowed to enter. Leaving the *# Decimal Digits* text box blank will allow the data abstractor to place the decimal anywhere within the number. Entering 0 (zero) in the *# Decimal Digits* text box will prevent the data abstractor from entering a decimal place.

During entry of a number, the data abstractor will be allowed to enter only one decimal place at any location within the number.

The user must also select at least two units from a group of units listed. If a new unit group from which to select is needed, use the Edit Units option under the UTILITIES menu to define it. The first unit in the list serves as the base unit in which the data are saved. For example, specifying Pounds and Kilograms as the two units will save the data in Pounds. Specifying Kilograms and Pounds as the two units will save the data in Kilograms.

If the user also selects "UTD" as a unit, the data abstractor may enter a number with unknown units. When the value is saved, a "U" is appended to the end of the string.

A feature "Require all digits" can be selected when creating this type of variable to force entry of all of the digits.

#### EXAMPLE:

Use variable type Number with units when the data abstractor is to enter a patient's height in centimeters or inches, and have the value converted/saved to centimeters.

**NUMBER SET.** A number set requires the user to specify the maximum number of whole digits and decimal place digits that the data abstractor is allowed to enter. In addition, the user can define an unlimited number of options that will appear on the screen. It is recommended that this not exceed more than 25 options.

A number set is a set of options (typically units of measurement) where a number can be entered for each option in the list. For example, the user might need to collect the number of years, months and days since a patient was last pregnant. In this case, the options listed would be years, months and days.

When the data are saved, each value is saved (in the order of the options) in a string separated by a semicolon. For example, 3 years, 2 months and 1 day would be saved as "3;2;1;".

A feature "Require all digits" can be selected when creating this type of variable to force entry of all of the digits.



#### Advanced User: Maximum Number Of Options

Since the user is limited to saving 254 characters of data, the maximum number of digits (whole plus decimal plus 1[for the semicolon]) multiplied by the number of options must be less than 254 characters. If it exceeds 254, a warning will be displayed.

**ICD9 CODE.** This variable type provides the data abstractor with ICD-9-CM diagnosis codes (and descriptions) and ICD-9-CM procedure codes (and descriptions) from which a selection(s) can be made. During the design, select the type of code (i.e. ICD-9-CM diagnosis or ICD-9-CM procedure codes) to be collected during data entry.

The user also has the option of saving the decimal point associated with each ICD9 code. To select this feature click on the Include *Decimal Point for Code* check box located beneath the choices for the ICD9 variable. A code originally saved as 30089 in the datastore database will now be saved as 300.89. If during the design process the option to include a decimal point associated with an ICD9 code is selected, it will be displayed on the **Data Entry** screen after a selection has been made.

**DIAGRAM.** This variable type allows the data abstractor to mark up one or more diagrams each identified by the diagram title. The user must add one or more diagram titles for the diagrams to be collected in the list provided. In addition, for each diagram, identify the bitmap file that is to be used.

Since the user can also enter remarks about the diagram and these data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

During data entry, the data abstractor will be able to mark up the bitmap for each case with lines, circles, arrows and text. In addition, he/she will be able to write comments for each of the diagrams.

**NOTE:**

Since this is a free form text field, the data collected will not be conducive for analysis.

**LABEL.** This variable type does not allow data entry. It is used to place a set of text on the screen that is not associated with any data entry. The user is not required to enter any information except the variable name, short title and screen title. Data are not saved for this variable type.



**Advanced User: Not Sure Of Variable Type**

If designing a variable without knowing its type, create a label as a placeholder and then later refine the variable by switching it to the applicable type.

## Single/Grid Variable Options

---

**MANDATORY.** When a variable is marked as Mandatory, on the **Post Processor** screen (if the designer has turned on post processing after case entry and/or post processing after screen entry) an error message will be given (if no entry exists for the variable) that the variable is mandatory and must be entered before the case can be marked as complete.

**NOTE:**

The default for each variable is always mandatory. The mandatory flag applies the same way to all of the duplicate variables.

**READ ONLY.** When a variable is marked as **Read Only**, the data abstractor will not be able to enter any data for that variable on the given screen during data entry. When creating a Read Only variable, remember to check off the Mandatory option. If the Mandatory option is checked on, the data abstractor will not be able to complete a case (if post processing after a case is turned on) because the Read Only variable, that is also mandatory, will not accept data entry.

**EXAMPLE:**

If the data abstractor is to enter the variable **Discharge date** on the **Demographics** screen, but only view this variable on the **Discharge** screen, mark the variable **Discharge date** on the **Discharge screen** as **Read Only**. (Hint: remember, the same variable can be used on more than one screen).

**DERIVED.** Performing an arithmetic or logical operation on another variable to derive a value creates a derived variable. Age and length of hospital stay are examples of derived variables. Age is derived by calculating the difference between the date of birth and the date of some event, often the admission date. Calculating the difference between the admission and discharge dates derives length of stay.

To begin, check the Derived option from the *Variable Options* box and create a variable of the type in which the derived value is to be saved. For example, **Age** would be a variable type Number (No Units). Since all derived variables are **Read Only**, this option is automatically selected by the system and cannot be changed. After creating the derived variable, edit the derived equation by exiting the screen and selecting the **Edit Rules** mode. Clicking anywhere on the derived variable using the left mouse button will produce a popup menu that contains the following rule types:

- Exit Case If



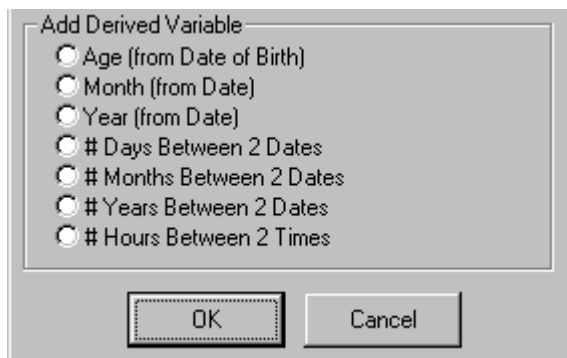
- Post Processor Warning If
- Post Processor Mandatory Override If
- Save ? To Me (Always)
- Save ? To Me If
- Delete Derived Equation(s) For Me

To create the derived equation, use the "Save ? To Me (Always)" or "Save ? To Me If" rules. The user can only apply one or the other "Save ? To Me" rule at a time to each variable type.

The **Save ? To Me (Always)** rule applies to variable types Date, Time, Date/Time, Number (No Units), Number (Units), and Number Set. Under all conditions, save a certain value, derived either by performing a calculation or assigning a value.

The **Save ? To Me If** rule applies to variable types Option (Pick One), Option (Pick One or More), Option (External List), Date, Time, Date/Time, String, Memo, Memo (ICD9), Memo (List Of), Number (No Units), Number (Units), Number Set, ICD9 Code, and Diagram. The **Save ? To Me If** rule allows different values to be saved to a derived variable under different conditions. Create a different rule that includes the criteria for each value to be saved.

Additional functions available to create numeric derived variables include:

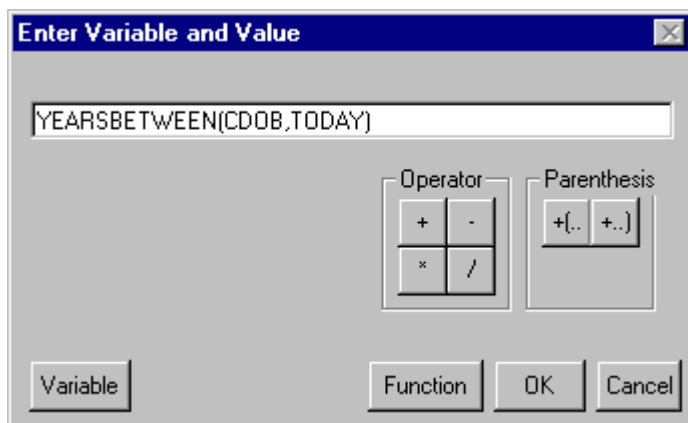


**Age (from Date of Birth).** Calculates age in years. Subtracts the date of birth from Today's date, then rounds the number to the lowest whole digit.

**Month (from date).** Extracts the numeric months (1-12) from a date and stores it as a derived variable.

**Year (from date).** Extracts the year from a date value that has been entered. If the length of the field of the derived value is shorter than the year that's been entered, the right portion of the year will be truncated.

**# ? Between 2 Dates.** Extracts the number of days, months, and years between two dates or the number of hours between two times.



#### NOTE:

When deriving a Date/Time variable, the user can only calculate using whole hours. For example, to calculate two days and one hour, use 49 hours. Calculate a Date variable using whole days and a Time variable using minutes.

**DEFAULT VALUE.** When a variable is marked as **Default Value**, a default value for that variable must be specified. If the variable is marked **Mandatory**, the data abstractor will be able to change the default value during data entry. If the variable is marked **Read Only**, the data abstractor will not be able to change the default value. The default values are loaded when a new case is created or when a record is added to a grid.

#### EXAMPLE:

By creating a date variable and choosing the Default Value option, the system will be required to save a date (e.g., 01/01/96) value for that variable.

**NO UTD.** When this feature is selected, the data abstractor will not be able to enter the "UTD" option (<SHIFT> + ?) for a variable. This feature applies to variable types: Option (External List), Date, Time, Date/Time, String, Memo, Memo (ICD9), Memo (list of), Number (no units), Number (units), ICD9 Code, and Diagram.

**INVISIBLE.** When this function is selected, only the variable's screen title will be displayed during data entry. This function is available for all variable types.

**RUN SAVE RULE ON DATA CHANGE.** When this function is selected, a variable associated with a "After Entry of Me Save ? to ? If..." rule will display a change in value each time the action variable's data is changed. By default, the value of the variable will change only after the action variable loses the focus.

This function can be used in combination with "Save To" rule type to create complex inter-relations between variables on the screen.

**NO ADD BUTTON (GRID VARIABLE OPTION).** When this function is selected, the <ADD> button will be invisible on the **Medications** or **Incident** screen during data entry. Selecting the No Add Button option prevents users from entering any data. Note that if the variable is marked mandatory and post processing after case is turned on, the user will not be able to complete a case. When the No Add Button option is used in conjunction with the Preload option, this ensures that only the predefined number of rows will be added during data entry.

**NOTE:**

Using the No Add Button and the No Delete Button options together with the **Preload** function, will ensure that the data abstractor can complete data entry for only the specified number of rows.

**NO DELETE BUTTON (GRID VARIABLE OPTION).** When this function is selected, the <DELETE> button will be invisible on the **Medications** or **Incident** screen during data entry. Selecting the No Delete Button option prevents users from deleting a row of data. This feature is useful when using the Preload option for the incident variable. By loading a specified number of options during data entry and having this option enabled, the user cannot delete any of the preloaded rows.

**NOTE:**

Using the No Add Button and the No Delete Button options together with the **Preload** function, will ensure that the data abstractor can complete data entry for only the specified number of rows.

**MAXIMUM # ROWS ALLOWED (GRID VARIABLE OPTION).** To enable this function (available for Medication and Incident variables), specify the number of rows a data abstractor will be allowed to enter during data entry in the *Maximum # Rows Allowed* check box.

**PRELOAD FROM VARIABLE OR PRELOAD # OF RECORDS (INCIDENT GRID VARIABLE OPTION ONLY).** To enable the **Preload From Variable** function, create at least one Option (pick one) variable, select this variable type from the grid, and click on the <PRELOAD VARIABLE=> button. The number of rows of data preloaded during data entry will match the number of options available from the variable (Option (pick one)).

To enable the **Preload by # of Records** function, enter the number of records to preload in the *Preload # of Records* text box. The number of rows of data preloaded during data entry will correspond to the number of rows entered here.

To clear either of the **Preload** functions, click on the <CLEAR PRELOAD> button.

## Popup Options

---

**MEDICATIONS SEARCH AID.** To enable this function, select this option from the POPUP OPTIONS menu item on the **Add/Edit Single Variable** screen. A popup box will appear from which the designer can select a specific medication category(ies).

During data entry, the abstractor can click the right mouse button on the variable and choose the Medications Search Aid option. A popup box enables the abstractor to search for a specific medication by typing in a string and clicking on the <SEARCH> button or by clicking on the <SEARCH FOR ALL> button.

**Edit Medication Search Aid**

Enter medication to search for in medications database per given categorie(s)

Enter medication string to search for (enter to search)

hepar

HEPARIN  
 HEPARIN IV  
 HEPARIN BOLUS  
 HEPARIN CALCIUM  
 HEPARIN D5W  
 HEPARIN DRDIP  
 HEPARIN DRIP  
 HEPARIN FLUSH  
 HEPARIN GTT  
 HEPARIN IN D5W  
 HEPARIN IN NS  
 HEPARIN IN 0.45% NACL 950U/HR  
 HEPARIN INFUSION  
 HEPARIN IV  
 HEPARIN LOCK

**FEET/INCHE CALCULATOR.** To enable this function, select this option from the POPUP OPTIONS menu item on the **Add/Edit Single Variable** screen. During data entry, the abstractor can click the right mouse button on the variable and choose the Feet-Inches Calculator option. A popup box enables the abstractor to enter the feet/inches and calculate the total inches, or enter the inches to calculate the total feet/inches.

**Edit Medication Search Aid**

Enter Feet and Inches to convert to inches

Feet/Inch Calculator

Feet

Inches

Converts to a total of

Inches

## Additional Features

---

**ALIGN NEW VARIABLES.** If this option is checked in the DESIGN menu, the user will receive a message asking to realign the variables on this screen. Answer yes to enable this option. Once this option is enabled, any time a new variable is added, the variables will automatically realign. Variables copied or moved to a screen using the **Copy/Move Variables to Screen** mode will still need to be realigned using the **Align Variables** or the **Size/Move Variable** functions. If this option is not selected, the newly added variables will continue to line up one on top of the other until they are aligned using the **Align Variables** or the **Size/Move Variable** functions. Note that by using this menu option, the user will always be aware of the space available on a screen.

**MEDICATION CODES THAT ENABLE A VARIABLE.** If the variable being entered is for a grid medication variable, the user will be prompted to identify the type of medication that will enable the variable being designed. Select the medication code(s) that will enable the variable during the entry of the medication. Use the **Edit Old MedQuest Medications Database** function in the UTILITIES menu to add and change the medication codes in the old MedQuest Medications database. If no medication type is selected from the list, all medications will enable the variable. The option to enable the variable on Valid Medications Only (defined during **Grid Design** for the medication being collected) can also be selected.

### EXAMPLE:

For a medication grid type, in the related variable's properties, identify the type of medication that will enable the variable. For example, to collect the start date for antibiotics and the start date and start time for ACE inhibitors, select the medication code associated with that type of drug for each of the variables added to the grid.

An option to "Enable on valid medications only" can also be selected. The related variable will only enable on the medication associated with the code(s) selected for the medication grid variable. If the medication grid variable does not have a specific code(s) selected, the related variable will never be enabled. This option should only be used to enable the related variable on the medication grid variable code(s).

**EDIT HELP.** Add/edit the help that will appear for the variable during data entry. For more details, see the section "Clinical Help" where editing help is described.

**QUERY/COPY.** To copy a variable and its properties from another module, use the Query/Copy function. This function uses the standard MedQuest query tool to identify and select a variable to copy. The query tool enables the user to identify a variable by selecting a screen, by searching the dictionary for a string of text, and by searching for variables that have a given keyword associated with them.

### NOTE:

Although the **Query/Copy** function can be used to copy a variable within the current module, it is recommended that the **Quick Copy** function be used instead. The **Quick Copy** function, which is accessed during the **Add Variable** or **Edit Variable** functions, allows the user to edit the variable being copied. The **Query/Copy** function will simply copy a selected variable to the screen on which the user is working.

### NOTE:

To copy a variable that exists on another screen with the same attributes as the selected variable yet not create a duplicate variable, use the **Quick Copy** function. The name of the variable can be changed while in the **Quick Copy** mode. By changing the name, it becomes a unique variable that will collect data independently of the variable from which it was copied.



### Advanced User: Data Dictionary Integrity Across Modules

When copying a variable from another module, be careful to plan the use of that variable. For example, if there are no changes to the variable, then keep the same variable name.

#### EXAMPLE:

If the variable **High blood pressure** (FIBBPHI) is copied from the Atrial Fibrillation module into the Congestive Heart Failure module and no changes are made to it, the variable name can remain as is. This will allow for analysis across two modules for the same variable. This is the most cost-effective way to "borrow" work that has already been done.

If a variable changes, it will not automatically be changed in the other modules where it is contained. If a "significant" change is made to a variable definition that would impact the data entry, then change the name of the variable to a new name. In this way the integrity of the old variable will be maintained.

**ISO 11179 STANDARD ATTRIBUTES ARE INCLUDED WITH MEDQUEST.** To enhance the MedQuest attributes by including the ISO standards (Basic attributes of data elements), click on the <ISO> button on the **Add/Edit Single/Grid/Incident Variable** screen. A list box containing the ISO standards will be displayed. By default, several of the ISO attributes will already be completed from entries made in MedQuest. In addition, there are some ISO attributes (such as the synonym and related context attributes) that the user can input. Refer to the ISO 11179 Standard for guidelines on how to fill in the attributes that are available to the user.

Select an attribute and click on the <VIEW/EDIT> button to view only or modify the attribute; click on the <HELP> button to display the help for each attribute; or click on the <CLOSE> button to exit the ISO attributes.

If an attribute is chosen that cannot be edited, a **View (Only) ISO Attribute #** screen will be displayed. If an attribute is chosen that can be edited, a **View/Edit ISO Attributes** screen will be displayed.

The user can edit the attribute by entering text into the text box. To save the edits, click on the <OK> button. To cancel the edits, click on the <CANCEL> button. From this screen the user can also view the help for this attribute by clicking on the <HELP> button.

#### NOTE:

The synonym and context attributes are paired together. If one or the other is chosen, the **View/Edit ISO Attributes** screen will display two separate text boxes to enter information for each of the attributes.

The <ADD> and <DELETE> buttons provide the user with the ability to add one or more items for each attribute. Click on the <ADD> button to display Item 1 and type in the first text box for the Synonymous Name attribute or the second text box for the Context attribute. Continuing adding an item and entering text for one or the other attributes as necessary. To delete an item, highlight an item and click on the <DELETE> button.

**SAVING CHANGES <OK>.** After completing the design of a variable, save the changes by clicking on the <SAVE> button. If there is any missing information or a change has been made that is not allowed, the user will be prompted by a message. Make any changes requested or answer any questions provided.

**CANCEL CHANGES <CANCEL>.** To cancel any changes made to a variable, select the <CANCEL> button. All changes will be ignored.

## Add a Grid Variable

---

There is a special type of variable called a grid variable. This type of variable allows the design of a "grid" or "spreadsheet" of variables to be collected. The grid variable can be described as a multiple entry variable because the user will be allowed to enter more than one record. For example, if the requirements specify collecting a list of medications, the designer would use this type of variable.

## Variable Types

---

There are two types of grid variables: medication and incident. The medication type allows for selection of a medication from a specified Medications database. The incident type acts as a counter for the record in the grid. For example, with the medication type enter Aspirin, Heparin, and ASA medications (3 multiple records). With the incident, add incident 01, incident 02, and incident 03 (3 multiple records) as a counter. For the medication type unlimited entries are allowed. For the incident type, up to 9999 multiple records are allowed.

### NOTE:

Be careful when selecting a grid variable type since the two types are usually used for distinctly different purposes.

**NAME.** Enter a unique name (32 characters or less) to identify each grid variable. The system allows reusing a grid variable that has already been defined (duplicate variable) on different data entry screens. By typing in the name of a grid variable that has already been used, a message will appear asking whether to retrieve that variable and its properties. The **Quick Copy** feature can also be used to duplicate an existing grid variable in the current module. This feature requires selecting a screen set, screen, and the variable to quick copy.

When reusing a grid variable, all of the other variables related to the selected grid are also reused.

### NOTE:

The capability of giving a variable a unique name using up to 32 characters is available for modules created in MedQuest 4.10 and later. The variable names in pre MedQuest 4.10 modules cannot be edited to contain more than eight characters.

**SHORT TITLE.** The short title (30 characters or less) is a short description of the grid variable and is used as a column heading on the grid. The grid variable is always the first column in the grid.

**SCREEN TITLE.** The long title (254 characters or less) is the long description of the grid variable.

## Mandatory

---

If a grid variable is marked as mandatory, the data abstractor will be required to enter at least one record during data entry or he/she will not be allowed to complete the case (if post processing is turned on).

## No Add Button

---

When this function is selected, the <ADD> button will be invisible on the **Medications** or **Incident** screen during data entry. Selecting the No Add Button option prevents users from entering any data. Note that if the variable is marked as mandatory and post processing after case is turned on, the user will not be able to complete the case.

## ***No Delete Button***

---

When this function is selected, the <DELETE> button will be invisible on the **Medications** or **Incident** screen during data entry. Selecting the No Delete Button option prevents users from deleting a row of data.

## ***Maximum # Rows Allowed***

---

To enable this function, specify the number of rows in the *Maximum # Rows Allowed* check box that a data abstractor will be allowed to enter.

## ***Medication Entry Criteria***

---

When defining a medication multiple entry type variable, the user may specify a set of criteria to limit data entry to one or more groups of medications. There are two methods available to accomplish this.

### **LIMITING ENTRY TO PREDEFINED CATEGORIES**

---

If the developer wants to limit the user to entering only a certain type of medication, he/she can identify this by selecting a type(s) from the *Valid Medication Types to Collect*: list box.

### **BUILDING A CUSTOMIZED MEDICATIONS LIST**

---

This option allows the developer to refine a list of medications to limit data entry to one or more categories of medications. Select the Build Customized Meds List option from the **Add Grid Variable** window and click on the <EDIT LIST> button. When defining a custom list, the developer has the option of selecting medications from either the old MedQuest Medications Database or the MediSource™ Lexicon Database, or both databases. When using this option, only the medications included in the custom list can be added during data entry.

To build a custom list, select the database from which to build the list. Enter a string in the search box and click on the <SEARCH> button. If there are any items to list, they will be displayed in the *Medications* list box. To add the search string to the custom list, click on the <INCLUDE IN LIST> button. To add an item(s) from the list, use the mouse to highlight one or more items or click on the <SELECT ALL> button to highlight all of the items. Click on the <ADD TO CUSTOM LIST> button to display the items in the *Custom List of Medications* text box. To deselect all of the medications in the search list, click on the <DESELECT ALL> button. To clear all of the medications in the search list, click on the <CLEAR ALL> button.

The user can also select one or more categories of medications and click on the <DISPLAY MEDICATIONS FOR SELECTED CATEGORY> button. The user can then use the functionality described above to create a custom medication list.

To remove one or more items from the custom list, highlight the medication(s) in the *Custom List of Medications* list box and click on the <REMOVE FROM CUSTOM LIST> button. To clear all of the medications from the custom list, click on the <CLEAR ALL> button.

### **NOTE:**

If a customized list is not created and a specific code is not selected, the data abstractor will be allowed to enter any type of medication.



## **PREVENTING DUPLICATE ENTRY OF MEDICATIONS**

---

The developer can prevent the data abstractor from entering duplicate generic medications. For example, if the user enters "**ASA**" (generic Aspirin) first, and then tries to enter "**Bayer**" (generic Aspirin) next, he/she would be prevented from entering the "**Bayer**" if the Prevent Duplicate Generic Names option was selected.

The developer can also enable the data abstractor to enter duplicate generics after bypassing a duplicate warning message. Select the option to "Warn on duplicate generic names" to accomplish this.

Finally, an option that allows the data abstractor to enter duplicate generic names without any warning can be selected.

## ***Incident Entry Criteria***

---

When an incident multiple entry type is selected, there is a set of criteria to answer. The designer can identify the key-link field(s) that will be used during the MedQuest Quality (IQC) program to match the different incident records.

In addition, the designer also has the option of preventing duplicate key-link incident records as well as limiting the total number of incident records that can be created by the abstractor (up to 9999).

### **NOTE:**

If the designer has specified not to accept duplicate key-link records, when the abstractor enters records with duplicate key-link fields, a warning message appears "This entry will result in a duplicate row. Accept duplicate?" The abstractor then has the option of accepting duplicate key-link incident records.

If, during design, a variable type Option (Pick One) has been added, the ability to preload the options during data entry is available. Highlight the variable to be preloaded (it must be a variable type Option (Pick One)), and click on the <PRELOAD=> button. The number of options defined in the variable will determine the number of records that are added during data entry.

### **NOTE:**

The **Preload** function can be used in conjunction with the No Add Button and No Delete Button options to require the data abstractor to complete data entry only for the rows created by the **Preload** function.

### **NOTE:**

The **Preload** function will override the Maximum # Rows Allowed option. For example, if the Option (Pick One) variable has seven options and the number of rows allowed during data entry is limited to five, seven rows will be added during data entry. However, the data abstractor will not be allowed to enter any additional rows.

## ***Add Variable (To Grid)***

---

Additional questions may be answered for each record entered in the grid. For the incident type variable, this is mandatory. For the medication type variable, additional related variables may be added to the grid.

### EXAMPLE:

To collect the surgical episodes that a patient experienced during a given hospital stay, use the incident grid variable (counter) and add two related variables--the surgery type (option pick one) and the date of surgery (date).

### EXAMPLE:

To collect all the antibiotics and ACE inhibitors given during the hospital stay and, for each medication, the date and time it was administered, use the medication grid type (with an antibiotics and an ACE inhibitors medication codes selected in the *Medications Entry Criteria* box). Then add related variables--the date of administration (date) and the time of administration (time).

For a medication grid type, when entering the related variable's properties, a prompt will appear to identify the type of medication that will enable the variable. For example, to collect the start date for antibiotics, and the start date and start time for ACE inhibitors, select the medication code associated with that type of drug for each of the variables added to the grid.

An option to "Enable on valid medications only" can also be selected. The related variable will only enable on the medication code(s) selected for the medication grid variable. If the medication grid variable does not have a specific code(s) selected, the related variable will never be enabled. Only use this option when enabling the related variable on the medication grid variable code(s).

### NOTE:

When performing any of the functions on a related variable (i.e., add, edit, resize, delete, move, create edit rules), the grid variable must be completely specified because it will be saved before the function is performed. The option of canceling these functions in a grid variable is not available as in a single variable. This occurs because the related variables must be linked to the grid variable for relational integrity.

When adding a related variable, the related variable is added as another column to the grid. The **Design** window that pops up is the same as the **Add Single Variable** window described in the Add a Variable section. Design the variable following the same rules as described in that section. The short title entered will appear in the column heading.

### ***Edit Variable (On Grid)***

---

To edit a related variable, select the column on the grid to edit and then select the **Edit** function. Note, the first column on the grid cannot be edited since this is the incident or medication grid variable and its data can be edited directly on the window. The help for related variables from this window can also be added/edited.

### ***Resize Variable (From Grid)***

---

To resize a related variable, select the column of the grid to resize and select this function. Note, the first column in the grid cannot be resized because this is the grid (incident or medication) variable.

### ***Resize Grid Columns***

---

To resize the grid columns to increase or decrease their widths, drag the border of a column to the desired size.

### **Delete Variable (From Grid)**

---

To delete a related variable, select the column of the grid to delete and select this function. Note, the first column in the grid cannot be deleted because this is the grid (incident or medication) variable.

### **Move Variable Left (On Grid)**

---

To move a related variable to the left, select the column to move on the grid (except the first column) and select the **Move Left** function.

### **Move Variable Right (On Grid)**

---

To move a related variable to the right, select the column to move on the grid (except the first column) and select the **Move Right** function.

### **Edit Variable Data Entry Rules (On Grid)**

---

To add or edit the data entry rule for a related variable, select a column on the grid in which to add the rule and select the **Edit Rules** function. When selecting this function, also select the rule type (currently enable, disable, and a warning after data entry). If a derived variable is being created, the rule types available include: Save ? to Me (Always); Save ? to Me If; and Delete Derived Equations for Me. The **Edit Rules** window can be used to create whatever logic is necessary.

#### **EXAMPLE:**

A medication has a related variable start date and the start date must be between the admission date and the discharge date. A warning rule can be created that warns the user if the date entered does not meet this criteria. After the user enters the start date for each medication, the warning will appear if the start date is not within the allowed range.

#### **EXAMPLE:**

For an incident with two related variables, **Procedures done** and **Types of procedures**, to enable **Types of procedures**, create a rule that **Types of procedures** will only be enabled if the **Procedures done** variable is answered "Yes."

#### **EXAMPLE:**

A variable can also be disabled in the same manner. To disable the **Types of procedures**, create a rule that the **Types of procedures** will be disabled if the **Procedures done** variable is answered "No" or "UTD."

### **Edit Variable Help (On Grid)**

---

This option provides the ability to add/edit the clinical help that will appear for the Medication/Incident variable (not the related variables) when the data abstractor performs data entry. When adding or editing a related variable, add/edit the help from the **Add/Edit Single Variable** window. (Note: a more detailed description of how help can be entered for a variable is provided in the Edit Clinical Help section.)

### **Exit (Save Grid Variable)**

---

When the properties of the grid variables are all entered, selecting this function can save them. If the required information has not been completed, a warning will appear and the save will not be allowed.

When the save is complete, the **Grid Design** window will be unloaded. Note that the grid variable is saved whenever a related variable is added to the grid.

## ***Cancel***

---

To cancel the changes made to the grid variables since the last save, select the <CANCEL> button.

### **NOTE:**

When performing any of the related variable functions (i.e., add, edit, delete), the grid variable is automatically saved.

## ***Edit Variable Mode***

---

The **Edit Variable** function switches the edit mode on or off. In variable edit mode, any variable in the design window can be edited by clicking on it. If the variable is a single variable (common type), edit the variable parameters in the same way as described in the Add a Variable section. If the variable is a grid variable, edit the variable parameters in the same way as described in the Add Variable (To Grid) section.

### **NOTE:**

The advantages of being in the Edit Variable mode is the ability to quickly select different variables to edit.

## ***Size/Move Variable Mode***

---

The variables on the screen are displayed in a What You See is What You Get (WYSIWYG) fashion. The system provides the ability to size the borders of each individual variable and to move each variable around on the screen. Switch the **Size/Move Variable** mode on or off by selecting that function.

When the **Size/Move Variable** mode is turned on, the user can click on any variable and, while holding down the mouse button, drag the variable to a new location. Releasing the mouse drops the variable in the new location.

In addition, the top and left locations of the variable can be entered (in Visual basic screen units) in the popup box provided or change the top and left values by using the up/down arrow buttons next to the top and left boxes.

The height and width of a variable can be changed by entering the new height and width into the boxes provided or by using the up/down arrow buttons next to those boxes.

Variables can also be sized by using the mouse. While the **Size/Move Variable** mode is on, place the cursor on the right side, right bottom corner or the bottom of the variable until the double arrow appears and drag the mouse to the desired variable size. Note that the variable cannot be resized from the left side or the top when using the mouse.

### **NOTE:**

Since the variables can now be adjusted on the screen, there is a responsibility to manage the real estate on the screen. It is possible to move variables around and hide variables beneath other ones. If this occurs, the variable on top of the hidden one must be moved to access the hidden one.

**NOTE:**

Since the variables can now be adjusted on the screen, there is a responsibility to manage the real estate on the screen. It is possible to move variables around and hide variables beneath other ones. If this occurs, the variable on top of the hidden one must be moved to access the hidden one.

**NOTE:**

The sizing of the variables is done differently for each variable type to optimize the performance speed of the system. Try experimenting with how the sizing works for each variable type before creating a lot of new variables of that type.

**NOTE:**

The size of a variable depends on the amount of information entered for that variable type. For example, creating a very long screen title will cause a variable to be resized to a larger size. This will prevent the placement of too many variables on a screen.

**NOTE:**

Windows 3.1 does not support multiple lines (wrapping) for option descriptions, so each option must be displayed as a single line on the screen. This does not apply to Windows 95 or Windows NT unless users are designing a module in which data is to be collected using Windows 3.1.

## ***Delete Variable Mode***

---

To specify a variable to be deleted, select the **Delete Variable** mode on the **Design** window. When deleting a variable, it is deleted from the screen on which it currently resides. When the deletion is finished, the variable will not appear in the **Design** window.

If the variable being deleted is only used on one screen (last reference to this variable), the system will ask whether to delete the contents of the variable as well. This means that all of the specific information that has been entered to define that variable will be deleted from the dictionary and it will not be recovered. This permanent deletion does not have to be accepted.

When a variable's contents are deleted, the data entry rules associated with that variable will also be deleted.



### **Advanced User: Deleting A Variable From The Master Table**

Since variables can be reused on different screens, deleting a variable will only delete it from the current screen. However, when that is the last reference to the variable, it will be deleted in its entirety. When this is completed, the variable is deleted from the **Master** table.

It is recommended that a variable only be deleted from the screen and not in its entirety. The variable may be needed to be used for historical purposes or as a template for creating another variable. When the variable is in the **Master** table, it still exists in the dictionary but does not have to appear on a screen and is not included for data entry or saved in the datastore database.



### Advanced User: Deleting A Variable From The Master Table With Data Entry Rules

In an expert data entry system there may be a rule associated with a variable that includes a deleted variable, e.g., a deleted variable may be included in a complex equation. This rule will execute as false since it cannot make reference on a screen to the deleted variable. It is the responsibility of the designer to manage the rules involving variables that have been deleted from a screen or the dictionary but are associated with the Action or Trigger fields of a rule.

## *Edit Variable Data Entry Rules Mode*

---

In the **Edit Rules** mode on the **Design** window, select a variable for adding a data entry rule. Depending on the type of variable, data entry or derived, a list of the available rule types will be displayed when selecting a variable.

**The rule types that are available for data entry variables include:**

- Enable Variable Rule
- Disable Variable Rule
- Skip to Variable Rule
- Hot Key Save Value to Variable Rule
- After Entry Warning Rule
- Save Value to Variable Rule
- Exit Case Rule
- Post Processor Warning Rule
- Post Processor Mandatory Override

**The rule types that are available for derived variables include:**

- Exit Case Rule
- Post Processor Warning Rule
- Post Processor Mandatory Override
- Save ? to Me (Always)
- Save ? to Me If
- Delete Derived Equation(s) for Me

Some of these rule types require specifying information in addition to the rule logic e.g., the enable variable rule requires specifying the variable that will enable it by selecting it from a list.

To add a new rule or edit an existing one, use the **Edit Rules** window. In this rule builder window, a general equation of logic can be constructed as a part of the data entry logic. The details for designing each rule type and using the rule builder to enter general logic are provided in the Data Entry Rules section.

## Align Variables

---

To have MedQuest align all of the variables on the screen in the order they were entered (or ordered them using the **Order/Indent/Unindent** function), select the **Align Variables** function. This will place the first variable in the upper left corner of the **Design** window and continue placing variables one below the other until the bottom of the window is reached. If there are still more variables after reaching the bottom of the screen, it will cycle back to the top of the screen and place the variable to the right of the last set.

After selecting the Align New Variables option in the DESIGN menu, a message will appear asking to realign the variables on this screen. Answer yes to enable this option. Once this option is enabled, any additional variables added to the screen using the **Add Variable** function will be aligned in the next available space. If this option is not selected, the newly added variables will continue to line up one on top of the other until they are aligned using the **Align Variables** or the **Size/Move Variable** functions.

### NOTE:

Variables copied or moved to a screen using the **Copy/Move Variables to Screen** mode will still need to be realigned using the **Align Variables** or the **Size/Move Variable** functions.

### NOTE:

This function uses the computer to calculate the top and left positions of each variable. Any top and left values entered using the **Size/Move Variables** function cannot be recovered because the newly aligned variables' values are saved to the database.

### NOTE:

If there are more variables than will fit on the screen, when the **Align** function reaches the bottom right of the **Design** window, it will place the next variable starting again at the top left. It is important to manage the real estate in the window to ensure that all of the variables fit on the screen.



### Advanced User: Use Of Add Variable And Align New Variables

The most advantageous use of the **Align New Variables** setting is to add a number of variables to the screen. Use the **Add Variable** function to continue to add variables one after the other, and the **Align** function will be performed after adding each variable. This saves having to add a variable, move it to a desired location, add a variable, move it, etc.

## Order/Indent/Unindent Variables

---

To view and change the order of the variables on a **Design** window, select this function. The **Order/Indent/Unindent Variables** function provides the capability to move the variables up and down in the list.

In addition, a hierarchy can be established for the variables by indenting them or unindenting them to show their relation to other variables on the **Design** window. This function establishes the relationship of one variable to another in a "parent-child" context.

Currently, the indentation is only used for two purposes. First, it is used for visual effect. For example, when producing a **Dictionary Survey** report, the indentation is used to visually reinforce that one variable is hierarchically related to another variable. Second, it is used as a part of the enable and disable rules. When a parent is disabled, all of its descendants will be disabled. This takes advantage of the hierarchy in creating "skip" patterns for answering questions during data entry.

The enable/disable rules can be added/edited in this function for any of the children variables; this makes it easy to build a set of rules as a part of the indented hierarchy. If an enable/disable rule is not added to an indented variable, that variable will always be disabled.



### Advanced User: Effective Use Of Enable/Disable Rules

The hierarchy created by indentation can be used to reduce the number of rules and improve the performance speed during data entry. This is especially true when the indentation is several levels deep (e.g., grandchildren, great grandchildren, etc.). However, it is not advisable to specify an enable/disable rule for each of a parent's descendants.

#### EXAMPLE:

There is a parent variable with one child and that child has 10 children (grandchildren). By using the indent feature, add one rule to the parent (to enable/disable the child) and then 10 rules to the child (to enable/disable the grandchildren). By not using the indent feature, 11 rules would have to be added to the parent (to enable/disable the child and the grandchildren) and 10 rules to the child (to enable/disable the grandchildren on entry of the child).

## Move an Indented Variable

---

When an indented variable is moved, it maintains its indentation. If the variable is moved out of its indented level, it will readjust its indentation to the level to which it is moved. This has implications on the construction of the enable/disable rules. By performing this type of function, "the implied hierarchy" of the variables is changing. If there are any enable/disable rules, they will still exist (they are not deleted) since it is assumed that the programming logic (i.e., rules) will have to be adjusted. It is advisable to build the indentation hierarchy before building any rules into the system to avoid unexpected results due to programming bugs.

## Changes to Indentation by Other Functions

---

The **Move/Copy Variables** function provides the capability to move or copy any of the variables on the current screen (**Design** window) to any other screen in the data entry system. The **Move** function will remove the reference to the variable from the current screen and place that reference on the new screen. The **Copy** function will copy the reference of the variable to the new screen and keep the existing reference intact.

## Move/Copy Variables to Screen

---

The **Move/Copy Variables** function provides the capability to move or copy any of the variables on the current screen (**Design** window) to any other screen in the data entry system. The **Move** function will remove the reference to the variable from the current screen and place that reference on the new screen. The **Copy** function will copy the reference of the variable to the new screen and keep the existing reference intact.





### Advanced User: Reuse Variables

The same variables can be used on different screens using the Copy function. If all of the screens have been designed, use the Copy function to copy references to the variable on one screen to all of the other screens where that reference is desired. Note, when using screen sets, this is quite common.

#### EXAMPLE:

There is a module with two screen sets (**History of MI** and **History of Atrial Fibrillation**). Design the **Current Findings** screen of the **MI** screen set first and add "high blood pressure." When this is done, copy that variable to the **Current Findings** screen for the **Atrial Fibrillation** screen set. The same variable is then reused across different screens.

## Delete All Variables

---

The **Delete All Variables** function operates in the same manner as the **Delete (Single) Variable** mode. The **Delete All Variables** function will remove references to all of the variables in the **Design** window, and the variables will not appear on the screen from which they were deleted.

In addition, the contents of the variables can also be removed from the dictionary. By choosing "No," the deleted variables will remain in the dictionary for reuse at a later time. By deleting the contents of the variables, all of the data entry rules associated with those variables will also be deleted.

## Query/Add Variables

---

This function can be used to retrieve variables from other modules and add them to the current **Design** window. It can also be used to query other variables in other modules (i.e., to quickly view the help for a variable) and to retrieve (add) help for another variable to the workspace. The objective of the general query interface is to provide a common access to variables in other modules.

**PATH TO LIST OF MODULES.** Point to any project file (which lists the modules and the locations of their dictionaries) by changing the directory path in the box provided. By default, the path is set to the current path used in MedQuest.

**SEARCH MECHANISMS (QUERY CRITERIA).** There are three ways a search can be performed for a variable(s). The first option, by Screen, lets the user select a module and a screen in that module. When the screen is selected, all of the variables on that screen are placed in the query result list.

Second, a variable can be searched by any word(s). This search can be performed on screen sets, screens or variables by selecting the type of query. To complete the query, select one or more modules to query, and then enter the word, string or phrase for the search into the boxes provided. Currently, the search can be done by name (variable only), short title, screen title, or help. These are the fields that are searched in the dictionary. Combinations of words can be entered that result in an "AND" search string. By selecting the **Query** function on the screen, the variables or screens that match the search criteria will be placed in the *Query Result* box.

Finally, a variable can be searched by keyword. These keywords are the same keywords that have been built using the **Edit Keywords** function in the utilities. This is accomplished by selecting the module to search and then typing in the keyword to search on. Note that as the keyword is being typed, an alphabetical search is performed and all keywords that match the string are displayed, plus the next 50 items in the keyword list. By selecting a keyword from the list and selecting the **Query** function, the variables that are classified by the keyword will be placed in the query result list.

**QUERY RESULTS.** Once the query has been run, the list of variables matching the query will be displayed in the *Query Results*: list box.

When selecting an item in the query result list, any help that has been entered for that item will appear in the help box provided. This makes it easy to see the help for other variables in other systems.

For variables in the query results list, additional details about a variable can be viewed by selecting the **Preview** function. This provides a more English-like description of the variable.

**ADD/COPY VARIABLES/HELP.** If the **Query Variables** window is being used to add variables to the current **Design** window, select all of the variables in the query results list that are to be added, and then select the **Add** function.

If the **Query Variables** window is being used to retrieve a variable's help into the current **Edit Variable Help** window, select one variable (only one variable is allowed to be selected) and then select the **Add** function.

If the **Query Variables** window is being used to copy a variable's properties to a variable in the **Design** window, select one variable and select the **Copy** function.

## ***Edit Screen Sets/Screens***

---

To build a data entry system, organize the questions (variables) that are to be answered. MedQuest enables these variables to be organized into screens or a set of screens (Screen Sets).

**SCREEN SETS.** A screen set is used to define a set of screens for which the information is to be collected together. For example, there may be a screen set that contains all of the data entry screens with questions about Myocardial Infarction.

Currently there is a limit of 20 screen sets. A new screen set can be added by selecting an existing screen set below which the new screen set will be placed and then selecting the **Add Screen Set** function. There will be a prompt to enter a unique screen set name and a description that will appear during **Data Entry**. The new screen set will appear on the list in the **Edit Screen Sets/Screens** window.

**SCREENS.** A screen can contain variables for data entry or can point to a set of subscreens that contain variables. Selecting the screen set or screen below which to add the new screen and selecting the **Add Screen** function can create a screen. There will be a prompt to enter a unique screen name and a description that will appear on the tab during data entry. There is a limit of 10 screens for each screen set.

**SUBSCREENS.** When there are more variables than can fit on one screen, it may be necessary to create a set of subscreens that are associated with that screen. Selecting the screen or subscreen below which to add the subscreen and selecting the **Add Subscreen** function can do this. There will be a prompt for the same information that appears when a screen is added. There is a limit of five subscreens for each screen.

**SCREEN CALL.** It may be necessary to use an existing screen more than once, e.g., the same **Demographics** screen may be used in all of the screen sets created for a given module. To reuse a screen, select the screen set or screen below which to add the screen call and then select the **Add Screen Call** function. Then select a screen from a list of unique screens currently used in the system and enter a description (which appears on the tab during data entry) for the screen call.

Note that all of the variables for the screen that is reused will appear on the screen wherever that same screen name is used. Also, if changes are made to that screen, these changes appear wherever that same screen name is reused.

**SUBSCREEN CALL.** The subscreen call works just like the screen call described above except that a screen (or subscreen) is being reused as a part of a subscreen.

It is important to emphasize that a screen, subscreen, screen call, and subscreen call are essentially the same; they provide a given set of variables for data entry. The objective in designing the collection of screens sets and screens is to construct the most efficient way for the data abstractor to enter data.

**MOVE UP/MOVE DOWN.** Screen sets, screens, subscreens, screen calls (which are screens) and subscreen calls (which are subscreens) can be moved up or down within their group by selecting the appropriate function. For example, one screen set can be moved above or below another screen set, but that screen set cannot be moved above or below a screen or subscreen. The same rule applies to screens and subscreens; they can be moved within their group of screens and subscreens, respectively.

**EDIT.** Use the **Edit** function to change either the unique name or the description of a screen set, screen, subscreen, screen call or subscreen call. It is convenient to change the descriptions but try to avoid changing the unique name for the component being used. Designers often start the process by uniquely identifying their components and then retaining that naming convention.

**DELETE.** Delete a screen set, screen, subscreen, screen call and subscreen call by selecting the item from the list and then selecting the **Delete** function. A screen that has subscreens cannot be deleted without first deleting the subscreens.

When deleting a screen set, screen, or subscreen (and its calls), no variables are deleted from the screen or subscreen (and its calls). The variables remain attached to the unique name in something that is called a "view." The only way to delete variables from a screen or subscreen (and its calls) is to use the **Delete Variable** or **Delete All Variables** function. These functions actually remove the variable from the "view" presented by the screen.



#### Advanced User: Recover A Lost Screen

If a screen has been deleted but that screen and the variables on that screen need to be recovered, use the **Add Screen Call** function to select the unique name of the screen to recover. In reality, the list of screens that appears when using the **Screen Call** function is a list of all of the screens (or "views") to which variables have previously been added.

**EDIT SCREEN RULES.** Rules can be added to screens and subscreens by selecting a screen/subscreen from the list and then selecting the **Edit Screen Rules** function. Three types of rules can be added; "Disable Screen/Subscreen If," "Do Not Load Screen/Subscreen If," and "On Entering Case Load Screenset ? If."

#### EXAMPLE:

The designer may want a particular screen completed (i.e., a **Prevalidation** screen) or a specific variable on that screen answered before the abstractor can proceed to another screen(s) (i.e., a **Demographics** screen). The designer would select the **Demographics** screen and add a rule stating not to load this screen if the **Prevalidation** screen is not equal to FILLED. Alternatively, the designer could select a variable on the **Prevalidation** screen such as Was vancomycin received as a discharge medication and state that if the answer is equal to NO, do not load the **Demographics** screen.

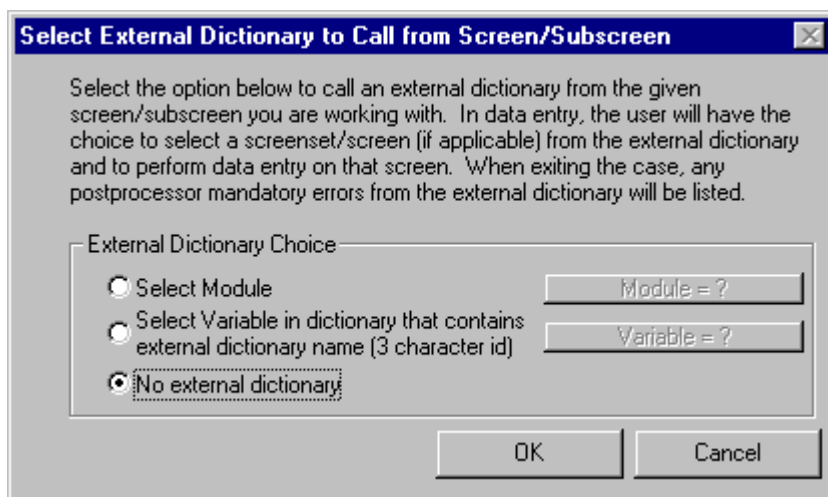
### EXAMPLE:

The designer may want the **Lab** screens to be disabled if the answer to the question "Any lab tests ordered" is "No."

### EXAMPLE:

The designer may want the **Patient** screen set to be loaded after data entry for the **Hospital** screen set has been completed.

**CALL EXTERNAL DICTIONARY.** An external dictionary can be called from a screen/subscreen by selecting a screen/subscreen from the list and then selecting the **Call External Dictionary** function. There is a choice to call a module or a string variable from the current dictionary. The default is "No External Dictionary."



If a module is selected, a popup box will appear from which an existing module can be selected. In data entry, the user will need to select the tab for the external dictionary call. He/she will then have the choice to select a screen set/screen (if applicable) from the external dictionary and perform data entry. The external dictionary screen will be loaded on "top" of the screen/subscreen selected for data entry.

To select a variable, first create the string variable that will contain the external dictionary name. Place the string variable, limited to a length of three characters (i.e., CCP = the name of the external dictionary), on a screen other than the screen from the external dictionary will be called. After creating this variable and selecting a screen, a popup box will appear from which the string variable can be selected.

In data entry, the user will need to enter the three character external dictionary ID into the string variable. The user can then click on the selected tab for the external dictionary call. He/she will have the choice to select a screen set/screen (if applicable) from the external dictionary and perform data entry.

### NOTE:

Since the external dictionary screen is loaded on top of the selected data entry screen, do not add variables to this screen during design. These variables will not be visible nor will the user have access to them during data entry. If they are mandatory, the abstractor will not be able to complete the case.

### NOTE:

When exiting the case, any post processor mandatory messages or warnings from the external dictionary will be listed. If there are any mandatory messages, the abstractor will not be able to complete the case.

### EXAMPLE:

The Minimum Data Set (MDS) module has external dictionary calls for state-specific screens. The user can complete a standard case for the MDS module and then complete the screen(s) for the state-specific information.

**EDIT HELP.** Help can be added to any of the items (i.e., screen set, screen, subscreen, screen call, subscreen call) by selecting the item from the list and then selecting the **Edit Help** function. This will open the **Edit Screen Help** window that is described in detail in the chapter on Clinical Help.

Note that the help for each item in the list is unique. If a screen is reused via screen calls, for example, the help will need to be entered for that reused screen; the help will not automatically be the same for the reused screen name.



#### Advanced User: Copy Help

If multiple screens and subscreens are reused, it may not be efficient to reenter the help for the duplicate screens. This can be avoided by using the **Query/Copy** feature on the **Edit Screen Help** window to minimize the effort. This feature allows the help for a given screen to be copied into the workspace so it can be copied to the new screen.

## Utilities

---

MedQuest contains a number of supplementary data sets that may be useful to modify or maintain. Following is a brief description of the design utilities currently available to enable this. (A more detailed description of their use is provided in the Utilities section.)

**EDIT MEDICATIONS.** For maintaining the old MedQuest Medications database.

**EDIT ABBREVIATIONS.** For maintaining the Abbreviations database.

**EDIT KEYWORDS.** For assigning (classifying) a variable to a keyword and adding/deleting keywords.

**EDIT UNITS.** For defining a group of units and their conversion factors that can be used when constructing a variable.

**EDIT HEADER VARIABLES.** If there are variables for downloading data but not placing on a data entry screen, this utility can be used e.g., use this utility to add/remove UB-92 or Breast Cancer Registry data variables to/from the data entry system.

When using a custom Opening screen with hierarchical objects, the user must add the unique identifying record to the list of header variables (see the section on Edit Opening Screen).

**EDIT DATA ENTRY SCREEN TOP.** This utility can be used to add variables that will appear at the top of the screen (along with their values entered) during data entry.

**EDIT VARIABLE ASSISTANT POPUP BOX.** If there are a number of variables and their values that are to be shown for read only reference by the user during data entry (e.g., display a series of dates and times), this can be done by adding and deleting variables using this option. These variables, along with their values entered, will appear during data entry when the Variable Assistant option is selected.

**EDIT CASE STATUS SCREEN.** To maintain the variables that appear on the case status screen after data entry, this utility enables the user to add new variables to (or modify existing variables in) the dictionary.

**EDIT DERIVED VARIABLES (NOT USED IN DATA ENTRY, ONLY ANALYSIS).** This function provides the capability to create variables or derived variables that can be used for analysis in the MedQuest Analyzer. This feature is very useful during the design process of a module. The developer and the analyst can work together to develop the data to be abstracted and create the derived variables needed to aid in the analysis process.

**EDIT OPENING SCREEN.** This function gives the user the capability to design an **Opening** screen. This is the screen that appears after the **Login** screen.

---

## Help

---

**GENERAL HELP.** The General Help provides general MedQuest system help.

**MODULE HELP.** If the clinical help entered in the module has been compiled, then it can be viewed (XXXDICT.HLP where XXX is the module acronym) here.

**GENERAL DATA ENTRY GUIDELINES.** If a help file explaining the general guidelines for data abstraction has been created, then that help file (DESGUIDE.HLP) can be viewed here. See the Clinical Help section for a brief description of how to create these guidelines.

**ABOUT MEDQUEST.** About MedQuest identifies the MedQuest and dictionary version information.

---

## CLINICAL HELP (DESIGN)

---

In MedQuest, clinical help can be added for the screen sets, screens, subscreens, and variables to the module being designed. This help is saved in the dictionary and can be used to build a Microsoft help file that can be retrieved and viewed during data entry. This help is entered in a structured format to maintain consistency among all of the modules.

---

## Create General Data Entry Guidelines

---

The module developer may use any commercial software available to create a help file describing the general guidelines used for data entry. Once this file has been created, name the file DESGUIDE.HLP and copy it into the MedQuest directory. The user can then access this file from three HELP menu locations in MedQuest: the **main Data Entry/Design** screen, and the main **Data Entry** and **Design** screens.

---

## Edit Screen Clinical Help

---

When editing help for a screen set, screen, or subscreen (and their calls), the user is presented with two basic sections into which help can be entered: sources and instructions. There are several assistive devices provided to paste clinical text that has already been defined into a specific section:

**SOURCES.** Lets the user select one or more standard medical record sources where the data abstractor is expected to find the data for the variables on the screen set, screen, or subscreen.

**HEADINGS (SOURCES).** Lets the user select some standard headings that might be useful to paste into the sources section.

**HEADINGS (INSTRUCTIONS).** Lets the user select some standard headings that might be useful to paste into the instructions section.

**BULLET.** Lets the user place a bullet in front of a set of words in the instructions section.

**Other features that are available include:**

**COMMENTS.** The user can enter comments about the screen set, screen, or subscreen for which help is being edited. These comments are separate from the help and can be included in the different reports where help is reported.

**QUERY/COPY.** To borrow help from another variable, screen, or subscreen use this option. This option uses the standard MedQuest Query window (described in the Design Functions section) to identify the help to copy. When retrieving help with a query, the help is retrieved into the boxes on the right side of the screen. To copy all of the retrieved help to the current help workspace, select the Paste All function. To copy a certain portion of the help, highlight that section with the mouse, and use the "<<" keys to copy just that highlighted section to the current workspace.

**PREVIEW.** To preview what the help will look like in data entry, select the Preview option. Note that this is the way help will look if it has not been compiled into a Microsoft help file.

Finally, save the help file changes by clicking on the <OK> button. The selection of the <CANCEL> button will cancel any additions/changes that have made to the help.

## Edit Variable Clinical Help

---

The help for a variable can be edited by selecting the **Edit Variable Clinical Help** function when adding or editing a variable or grid. The variable help can be edited in the Design option by clicking on a variable in the **Design** window with the right mouse button and selecting the Edit Help option.

When editing help for a variable, three basic sections are presented into which the help can be entered: question, sources, and instructions. Several assistive devices are provided for pasting clinical text that has already been defined into a specific section:

**TITLE.** Pastes the variable screen title into the question section.

**SOURCES.** Pastes one or more medical record sources into the sources section.

**HEADINGS (SOURCES).** Provides some standard headings that might be useful to paste into the sources section.

**HEADINGS (INSTRUCTIONS).** Provides some standard headings that might be useful to paste into the instructions section.

**BULLET.** Allows a bullet to be placed in front of a set of words in the instructions section.

**OPTIONS.** Pastes in all the options that have been listed for this variable (if it is a variable type Option (Pick one) or Option (Pick one or more)). For example, this will list all the options for **Discharge status** that have been entered for the variable. Note that this is only useful if the variable's option have already been fully defined.

**MEDICATION.** Used to select one or more medication codes and retrieve all of the drug names associated with the selected code(s) into the workspace. For example, use this device to list all of the antibiotics that are listed in the Medications database. Note that it will not be able to list all the



medications if the list is too long. In addition, it is impractical for a data entry user to scroll through page after page of medications that are required for entry. It is better to collect the medications using the grid type variable (variable type medication) and to specify within that design to collect only one type of medication (e.g., antibiotics).

Other features that are available include:

**COMMENTS.** Enter comments about the variable that is being edited. These comments are separate from the help and can be included in the different reports where help is reported.

**QUERY/COPY.** To borrow help from another variable, use this option. This option uses the standard MedQuest Query window (described in the Design Functions section) to identify the help to copy. When retrieving help with a query, the help is retrieved into the boxes on the right side of the screen. To copy all of the retrieved help to the current help workspace, select the Paste All function. To only copy a certain portion of the help, highlight that section with the mouse, and use the "<<" keys to copy just that highlighted section to the current workspace.

**PREVIEW.** For previewing the appearance of the help for data entry. Note that this is the way help will look if it is not compiled into a Microsoft help file.

Finally, save the help file changes by pressing the <OK> button. The selection of the <CANCEL> button will cancel any additions/changes that were made to the help.



#### Advanced User: Paste Help From A Word Processing Document

In a typical scenario, the help may be built in a word processing environment. When it is finished, cut/paste/copy the contents of the help into the sections provided in the **Edit Variable Help** window. All Windows applications provide the ability to cut/paste/copy highlighted text. The common keys for doing this are the <CTRL>+ <C> (copy), <CTRL>+ <V> (paste) and <CTRL>+ <X> (cut) keys. The copy/cut keys will copy the contents of text that has been highlighted to a buffer from which it can then be pasted using the <CTRL>+ <V> keys. See the systems manager for assistance on this Windows feature.

#### NOTE:

Be aware that if using the cut/paste/copy, only paste ASCII text into the **Edit Help Design** window. If there are any special codes that are not recognized by the Microsoft help compiler, the help will not be compiled. This can be accomplished by saving the file as an ASCII text file and then using cut/paste/copy from the text file with no special embedded word processor codes.

## Compile Clinical Help Reports

After entering the clinical help into the dictionary, it is saved just as if it was entered into a memo field. If the data abstractor is supposed to see the help just as it is in the Design option using the <PREVIEW> button, then the clinical help does not have to be compiled. However, a utility has been provided in the **View/Print Reports** function to compile the help into a Microsoft Help (.HLP) file. A Microsoft help file is a hypertext file that most Windows software products use to present their help. The advantage of the Windows compatible help file is that it is easier to read and to find information than a noncompiled version.

The help can be compiled using a help compiler from Microsoft such as the program HCP.EXE (version 3.10 or later). The compiler is usually distributed with different development tools available from Microsoft and other software vendors. (Contact Microsoft to obtain a copy of the required version.)



When the help is compiled, several things are done to improve the look of the help on the screen. First, the screens and variables are formatted with proper styles (i.e., Heading 1 and Heading 2). Next, any headings that were added using the heading assistive device are bolded. Finally, all of the bullets are placed in the locations that have been identified.

When the compiler is finished, the help file will automatically be used during data entry. This file will have the name: "XXXDICT.HLP" where XXX represents the module acronym (e.g., CCPDICT.HLP).

This file can also be copied and sent to other users who are interested in the clinical help.



### Advanced User: Clinical Help Does Not Compile

Sometimes the clinical help file will not compile. The most common reason for this is that pasting text from a word processor program that has special codes in it has corrupted the help saved in the dictionary. The help compiler does not know how to compile certain special characters and will not complete the compilation. Be careful not to copy any special word processor codes when pasting help from other sources into the **Edit Variable Help** window.

---

## DATA ENTRY RULES (DESIGN)

---

MedQuest can be used to program rules that will be applied during **Data Entry**. This gives the designer the ability to build an "expert" data entry system. Typical types of rules that might be built include warning the user when a value is not within a specified range or adding a rule to "skip" entry of a variable.

In the MedQuest **Design** process, there is the ability to add different types of rules depending on the "event" that takes place in **Data Entry** (e.g., during entry, after entry, on exiting the case, etc.). For each of these rules, the MedQuest **Rule Editor** can be used to build the logic statement (or condition) that to test. The details on how to use this **Rule Editor** are described at the end of this section.

---

## Edit Rules (Rule Editor)

---

The **Rule Editor** is the design tool that can be used to add and edit data entry rules. All of the logic in the rules will be built using the **Rule Editor**. There are a few terms that should be understood before beginning to create rules.

**RULE.** A set of logic that evaluates to true or false, is associated with a given event (for example, on entry of a variable), and whose result, if true, produces a specific action (e.g., disabling a variable). MedQuest can have many rules for many different events and for many different actions. In abstract terms, a rule can be described by the following: "On Event X Execute Rule Y. If Rule Y Is True Then Do Action Z."

**LOGIC.** Each row of logic must contain a left side(cell), operator, right side(cell), and (if not the last row of logic) an "AND" or an "OR" that ties the rows together. Together, these rows make up the rules logic.

**CELL/LEFT SIDE LOGIC/RIGHT SIDE LOGIC.** Each row of logic must have a left side (cell) and a right side (cell) which contain variables and/or values that can be evaluated. For example, both a left cell and a right cell can contain "XYZ" (a variable) or "(TEMP-32)\*10" (a variable and value), and a right cell can also contain "01/02/92" (a value).

**OPERATORS.** The left side logic and right side logic can be evaluated and their results compared to each other using a comparison operator to determine if each row is true or false.

**PARENTHESSES.** Different rows of logic can be grouped together into a complex logical expression.

**PARENT.** A variable whose value can enable or disable the data entry value of another variable. For example, if the response to the parent variable **Medications prescribed on arrival** is false, then the child variable **Date medications prescribed** cannot accept any data entry value.

**CHILD.** A variable whose data entry is enabled or disabled by the value of another variable. For example, the variable **Date medications prescribed** can accept a value only when the parent variable **Medications prescribed on arrival** is true.

## *Add/Delete/Edit a Rule*

---

There are several functions that can be used to add/delete/edit a rule.

### **Current Rules**

---

When the **Rule Editor** opens up the **Edit Rules** window, any rules that already exist are listed at the top of the window. The current rule is the one that is highlighted and whose English translation is displayed in the box at the top of the screen. The English translation of any rule can be viewed (and that rule loaded for editing) by selecting that rule from the list in the upper left corner of the window.

The logic for the current rule is displayed in the grid in the middle of the window. This is where the user can edit the logic for the rule.

### **Add a Rule**

---

To add a rule, select the **Add** function. This will clear the logic from the previous rule (and save it) and preload the grid with the "expected" logic for the new rule. The expected logic (which can be changed) is the **Rule Editor's** best guess as to the logic for the type of rule that is being added. For example, for a disable rule, the variable name for the parent variable will be placed in the left cell, and the operator will be changed to "=". This is because in most circumstances the user will be disabling the child variable if the parent is equal to some value. Different types of variables fill the logic grid in different ways.

### **Delete a Rule**

---

If a rule is deleted, the user will not be able to recover it; the rule is removed from the data dictionary.

### **Save a Rule**

---

There is the option to save a rule, as the module is being designed, to ensure that the rule does not get lost if an accident occurs, etc. Note that in addition to being able to explicitly save a rule, the **Rule Editor** will ask to save a rule whenever an action changes the current rule.

### **Order Rules**

---

The user can change the order in which the rules are executed by using the **Up** and **Down** functions. The last rule in the list is the last one executed and takes precedence over any of the previous rules' actions.

## Apply Always

---

This option should be selected when a rule needs to be executed upon loading a case. The logic for this rule is not limited to a specific screen, e.g., a Post Processor Rule would need to execute upon exiting a case or exiting a screen. The following three rules are Apply Always only: the Skip to Variable Rule, the Exit Case Rule, and the Post Processor Warning Rule. These three rules do not have the option of the Apply Only on Current Screen rule.

## Apply Only on Current Screen

---

This option should be selected when a rule only needs to be executed upon loading a screen. For example, for an Enable Rule related to a variable on the same screen, there is the option of applying this rule only on the current screen since its logic is limited to this screen. The rules that have the option of Apply Always or Apply Only on Current Screen include: the Enable Variable Rule, the Disable Variable Rule, the Hot Key Save Value to Variable Rule, the After Entry Warning Rule, the Save Value to Variable Rule, and the Post Processor Mandatory Override Rule.

## Retry Only

---

This option is only available for the After Entry Warning Rule. This gives the designer the option in data entry to enable the data abstractor to only enter a valid answer to a question (variable) or accept an answer that is not within the limits of the warning.

## Edit Rule Grid

---

Most of the work takes place in the edit rule grid. To add to or edit a cell, click on that cell. The **Rule Editor** requires a rule to be entered from left to right. Therefore, enter the left cell first and then the right cell.

Click on the left cell to display a popup menu that requires the user to enter a variable, a variable and value, a screen, or add or remove a left parenthesis. In the operator cell, select the operator by clicking on the arrow button that will display the available operators.

Click on the right cell to display a popup menu that requires the user to enter a variable, a variable and value, a value (date, number, etc.), a keyword value "Filled" or "Empty," add or remove a right parentheses, or select null. In some cases, the user may only be presented with a subset of these options due to the type of variable being used.

**VARIABLE.** When selecting a variable, the user will be asked to select a variable from the *Select Variable* popup box.

**VALUE.** When selecting to add a value, the user will be presented with a window that lets him/her enter a value specific to the type of variable that is in the left cell, e.g., if the variable in the left cell is an Option (pick one), then the options for that variable will be displayed.

**VARIABLE PLUS VALUE (EQUATION).** When selecting to add a variable plus a value, the **Equation Design** window will appear. The **Equation Design** window gives the user the capability to build a more complex set of logic for certain variables. Following are the types of equations that can be utilized for the different variable types:

- Date. Variable + or - days (e.g., Discharge date - 2 days)
- Time. Variable + or - minutes (e.g., Arrival time + 20 minutes)

- Date/Time. Variable + or - hours (e.g., Discharge date/time + 48 hours)
- Number (no units). Complex equation
- Number (units). Complex equation
- Number (set). Complex equation

**SCREEN.** Select a screen in the left cell to include in the logic for Do Not Load and Disable Screen rules. Select a variable or a variable and a value in the left cell to include in the logic for On Entering Case Load ? Screenset If rule.

**FILLED/EMPTY.** Select the word FILLED or EMPTY in the right cell when a screen has been selected in the left cell. A filled screen is one with no variables that are enabled, mandatory, or blank. An empty screen is one that has no values entered for any of the variables on the screen.

## Complex Equation

---

When building a complex equation, the user can build the string by entering any equation. Use the different assistive devices to add a variable, add operators and parentheses, and enter numeric values. When working with a date or a time, the user is limited to adding or subtracting the applicable time unit.

## Other Edit Rule Grid Functions

---

The other assistive devices in the **Rule Editor** can be used to clear the highlighted cell, row of logic, or all the logic, or preview the English or SQL translation of the logic that is currently being edited.

In addition, if the user is required to enter other information (i.e., the variable to enable or disable, or the value to save to a specific variable), then the user will be presented with a button or popup box to perform this function.

For the Post Processor Warning Rule and the After Entry of Me Warn If Rule, the user has the capability to edit the warning message. Click on the <EDIT MESSAGE> button to display a *Warning Message* text box in which to type a message.

## Select Rule Type

---

When designing a dictionary, the user has the ability to add data entry rules from a number of different places. These functions are made available where they are most appropriate. For example, after adding variables, switch to the **Edit Rules** mode, click on the variable for which to add the rule, and select a rule type to display the **Rule Editor**.

### EXAMPLE:

To add a warning that a number should be between 0 and 100, select the <EDIT RULES MODE> button and then click on the variable for which to enter the rule. A Rule Types popup box will be displayed. Select rule type "**After Entry of Me Warn If**" to build the rule.

### EXAMPLE:

To add a rule to disable (or skip) a variable which is a child of an indented parent, select the **Order/Indent/Unindent** function and then select the child (indented) variable to disable. Select the **Disable If** function. On this screen, the **Rule Editor** is smart enough to understand that the user wants to disable this variable when the parent value is changed.

Following are the locations where a rule for a variable can be added or edited:

- Design Window
  - Edit Enable Variable Rule
  - Edit Disable Variable Rule
  - Edit Skip to Variable Rule
  - Edit Hot Key Save Value to Variable Rule
  - Edit After Entry Warning Rule
  - Edit Save Value to Variable Rule
  - Edit Exit Case Rule
  - Edit Post Processor Warning Rule
  - Edit Post Processor Mandatory Override Rule
  - Edit Save ? to Me (Always) (Derived Equation)
  - Edit Save ? to Me If (Derived Equation)
  - Edit Delete Derived Equation(s) for Me (Derived Equation)
- Order/Indent/Unindent
  - Edit Enable Variable Rule
  - Edit Disable Variable Rule
- Grid Design Window
  - Edit After Entry Warning Rule
  - Edit Enable Variable Rule
  - Edit Disable Variable Rule
  - Edit Save ? to Me (Always) (Derived Equation)
  - Edit Save ? to Me If (Derived Equation)
  - Edit Delete Derived Equation(s) for Me (Derived Equation)
- Screen Sets/Screens Design Window
  - Edit Do Not Load Screen/Subscreen Rule
  - Edit Disable Screen/Subscreen Rule
  - On Entering Case Load ? Screenset If

Following is a description of each of the rule types that can be created.

## Edit Enable Variable Rule

---

Use this function to specify the logic that enables a given variable when the value of another variable is changed (i.e., on entry of variable A, enable variable B if rule is true). This is the type of rule used to create "skip" patterns. Add this rule type to a variable in the **Design** window, the **Grid Design** window, the **Order/Indent/Unindent** window, or by using the <EDIT RULES MODE> button.

After selecting the **Edit Rules Mode** function in the **Design** window or clicking on the <EDIT RULES MODE> button, select the variable to enable. The **Rule Editor** can then be used to add the logic. The logic entered in the **Rule Editor** can be any general equation, but when adding a new rule it assumes that the "parent" variable is included in the logic (it does not have to be included). The "parent" variable is the variable that was originally clicked on to create the rule in the **Design** window. During **Data Entry**, the rule will execute when the "parent" variable's value is changed. If the rule is true, then the "child" variable will be enabled; otherwise it will be disabled. This same process would also be used for the **Grid Design** window.

When selecting this rule type in the **Order/Indent/Unindent** window, a "child" variable must already be selected in the variable list on the screen. The **Rule Editor** can then be used. Since MedQuest can look at the variables and know what the "parent" variable is for the "child," the option will not be available to change the "parent" variable that will enable the "child."

### EXAMPLE:

Enable "MI Location" if "MI Present" equals "Yes." This can be done by moving "MI Location" beneath "MI Present" and then indenting "MI Location." Highlight the "MI Location" variable and select the On Entry of Me Enable ? If... option, and use the **Rule Editor** to enter the logic "If MI Present" equals "Yes."



### Advanced User: Optimum Programming Approach

It is recommended that the enable and disable rules be built using the **Order/Indent/Unindent** window (for single variables). The enable/disable rules are special rules that use the indentation level of a variable to determine whether the variable is enabled or disabled. All children variables (and other descendants) of a parent that is disabled will automatically be disabled. In this way, the hierarchy created by the use of indentation provides a convenient structure to minimize the logic that must be created. Be careful: if there is not an equation for enabling or disabling an indented variable, that variable will always be disabled.

## Edit Disable Variable Rule

---

The discussion for the enable variable rule also applies to the disable rule. The only difference is that if the rule is true then the "child" variable will be disabled; otherwise it will be enabled.

### EXAMPLE:

Disable "Last Systolic Blood Pressure" if the "Discharge Status" is "Expired." On the **Design** window, select the **Edit Variable Data Entry Rule** mode, click on the "Discharge Status" variable and select the "On Entry of Me Disable ? IF" rule type. Select the variable to disable by selecting "Last Systolic Blood Pressure" and then use the **Rule Editor** to enter logic for "If Discharge Status is Expired."



### Advanced User: Disable/Enable Variables On Other Screens

Variables (excluding grid variables) can now be disabled/enabled on other screens when entering values on a given screen. For example, a rule can be created to disable the **Lab values at discharge** variable on the **Labs** screen when the **Discharge Status** on the **Discharge** screen is "Expired."



### Advanced User: Cascade Disable/Enable Rules

Since it is now possible to program virtually any set of complex rules, it is also possible to introduce logic bugs into the data entry system. It is important to use good programming practices to build a complex system. Be especially wary of building too many disabling/enabling rules that reference the same variables in different ways; this tends to lead to cascading logic that can put the results of the rules into an unknown state. To avoid cascading, MedQuest will only execute the rules associated with a variable, and the last rule executed will take precedence (the rules can be ordered in the **Rule Editor**).

## Edit Skip To Variable Rule

---

This rule type is used similarly to the **Enable Rule** function to create "skip to" patterns. There are several advantages in using this rule type. The designer does not need to create a hierarchy level to enforce a skip to rule. Also, any variables between the action variable that has the skip to rule and the variable being skipped to will automatically be disabled. This gives the designer the ability to use fewer rules.

After selecting this rule type for the designated variable, the **Rule Editor** can be used to add the logic and to select the variable to skip to. The logic entered in the **Rule Editor** can be any general equation, but when adding a new rule it assumes that the "action" variable is included in the logic. The "action" variable is the variable that was originally clicked on to create the rule in the **Design** window. During **Data Entry**, the rule will execute when the "action" variable's value is changed. If the rule is true, then the variables between the "action" variable and the "skip to" variable will be disabled.

### NOTE:

The user cannot skip to a previous variable.



### Advanced User: Skip To Variables On Another Screen

The designer may want to collect data for two different types of procedures (i.e., Medical or Surgical procedures). The designer would create his/her data entry system with questions for each type of procedure on separate screens. Depending on the data abstractor's response to the variable **Was this a medical or surgical procedure**, he/she would skip to the screens associated with the specific procedure.

## Edit Hot Key Save Value To Variable Rule

---

This rule type is used to save a value from another variable to the one being entered when a hot key is pressed. For example, this rule type can be used to copy the admission date from a downloaded variable in the header. This might be done to assist with verifying the correct entry of that variable.

This rule can only be added from the **Design** window and the <EDIT RULES MODE> button. Select a variable and then select this rule type. The **Rule Editor** can then be used to enter the rule's logic. Enter the variable whose value is to be saved. Click on the <SAVE " TO> button to select this variable. In addition, the "Hot Key" can be used to invoke the save action.



## Edit After Entry Warning Rule

---

This rule type is used to build a warning message that appears after the data abstractor completes entry of a variable. This type of rule can be used to warn the data abstractor that a value should be between a specified upper and lower bound. A rule of this type can be added from both the **Design** window and from the **Grid Design** window.

When this rule is added from the **Design** window, select the variable to add an edit warning and then select this rule type. The **Rule Editor** can then be used to enter the rule's logic. After entering the rule's logic, edit the warning message. While in the **Rule Editor**, click on the <EDIT MESSAGE> button. A popup box will open. Type in the message then click on the <OK> button to save the message. To only display the user defined message, check the *Display Only Edited Message* check box. This will keep the default "machine language" from being displayed during data entry. In **Data Entry**, the rule is executed when the data abstractor finishes entry of a variable. If the rule is true, a warning message will be displayed.

These types of rules can also be added in the **Grid Design** window, e.g., add a warning that ensures the date a medication is administered is between the admission and discharge dates.

### EXAMPLE:

When using the **Hot Key Save Value to Variable** rule, the designer may want to add a warning message to alert the data abstractor if the data entered in a variable, i.e., **Social Security Number**, does not equal the corresponding downloaded value, i.e., for **Social Security Number**.

## Edit Save Value to Variable Rule

---

This rule type saves a value to a variable after completing entry of another variable, e.g. save a value of "0" to the variable **Lowest pulse** if the variable **Discharge status** is "Expired." This rule type can only be added from the **Design** window.

After pointing to the variable whose entry executes the rule and selecting this rule type, use the **Rule Editor** to select the variable to save a value and the value to save. The **Rule Editor** can also be used to enter the logic for the rule. The rule executes when the data abstractor finishes entering a value for the variable selected on the **Design** window.



### Advanced User: Save A Value To A Variable That Is Disabled

It is possible to save a value to a variable that is disabled through the enable/disable rules. MedQuest does not check for this state since it assumes that the logic is correct. MedQuest always saves the value if the rule result is true.

## Edit Exit Case Rule

---

Use this rule type to warn the data abstractor to stop entering data and exit the case, e.g., stop entry of a case if the variable **Discharge Status** is "Expired." This rule can only be added from the **Design** window.

Add this rule by selecting a variable whose entry will invoke the rule and then selecting the rule type. The **Rule Editor** can be used to enter the logic that will force the data abstractor to exit the case.

After entering the rule's logic, the user can edit the warning message. While in the **Rule Editor**, click on the <EDIT MESSAGE> button. A popup box will open and the user can type in a message. Click the



<OK> button to save the message. To only display the user defined message, check the *Display Only Edited Message* check box. This will keep the default "machine language" from being displayed during data entry.

All of the exit case rules are executed when the case is first opened and after data are entered for each variable. In this way, if any exit conditions exist, the data abstractor will not be allowed to continue with data entry.



#### Advanced User: Too Many Exit Rules

If there are too many exit rules, this will decrease the speed of the data entry system because all of the rules must execute after data are entered for each variable. Therefore, avoid adding too many exit rules to the system.

## Edit Post Processor Warning Rule

---

To create a general warning message that appears in the **Post Processor Warning** window (after entry of a screen or after entry of a case), select this rule type, e.g., to display a warning that the data abstractor should check the **High blood pressure** value because the answer to the variable **Hypertension** is "Yes." This rule type can only be added from the **Design** window.

Add this rule type by selecting the variable for the post processor warning and then selecting the rule type. Use the **Rule Editor** to enter the logic associated with the warning. After entering the rule's logic, the user can edit the warning message. While in the **Rule Editor**, click on the <EDIT MESSAGE> button. A popup box will open and the user can type in a message. Click the <OK> button to save the message. To only display the user defined message, check the *Display Only Edited Message* check box. This will keep the default "machine language" from being displayed during data entry.

## Edit Post Processor Mandatory Override Rule

---

To override a mandatory warning in post processing for a given variable (after entry of a screen or after entry of a case), select this rule type. This rule type can only be added from the **Design** window.

Add this rule type by selecting the variable for the post processor mandatory override and then selecting the rule type. Use the **Rule Editor** to enter the logic associated with the mandatory override.

#### EXAMPLE:

The rule indicates that if the **Discharge Disposition** variable is answered "Patient expired," do not answer the **Discharge BP** variable. By assigning the Post Processor Mandatory Override Rule to the **Discharge BP** variable, then when the answer to **Discharge Disposition** is "Patient expired," the **Discharge BP** does not require data entry.

## Edit Save ? to Me (Always) Rule (Derived Variable)

---

The **Save ? To Me (Always)** rule applies to variable types Date, Time, Date/Time, Number (No Units), Number (Units), and Number Set. Under all conditions, it saves a certain value derived either by performing a calculation or assigning a category. This rule type can be added from the **Design** window or the **Grid Design** window.

Add this rule type by selecting the variable for which a value is to be saved and then selecting the rule type. Use the **Rule Editor** to enter the logic associated with a derived value.

## Edit Save ? to Me If Rule (Derived Variable)

---

The **Save ? To Me If** rule applies to variable types Option (Pick One), Option (Pick One or More), Option (External List), Date, Time, Date/Time, String, Memo, Memo (ICD9), Memo (List Of), Number (No Units), Number (Units), Number Set, ICD9 Code, and Diagram. The **Save ? To Me If** rule allows different values to be saved to a derived variable under different conditions. The user can create a different rule that includes the criteria for each value to be saved. This rule type can be added from the **Design** window or the **Grid Design** window.

Add this rule type by selecting the variable for which a value is to be saved and then selecting the rule type. Use the **Rule Editor** to enter the logic associated with a derived value.

## Edit Delete Derived Equation(s) for Me

---

When this function is selected, it will delete all of the equations associated with a derived variable. This rule type can be added from the **Design** window or the **Grid Design** window.

Invoke this rule type by selecting the variable for which a value is to be saved and then selecting the rule type. A message will appear asking to delete all the derived equations associated with this variable. Click "Yes" to delete the equations or "No" to cancel.

## Edit Do Not Load Screen/Subscreen Rule

---

To prevent a user from entering data on a given screen until certain conditions are met, use the **Do Not Load Screen** rule. The user can add these rules to any screen, subscreen or screen call from the **Screen Design** window.

Typically this rule is used to require the abstractor to enter data on screens in a certain order or to insure that he/she will not enter data on other screens until certain key screens are filled in first. With this rule type, the user can also include logic on whether a screen is empty (screen has no data entered) or filled (screen has no mandatory variables that are enabled and without a value). Note, this rule type cannot be used with **Sequential Data Entry**.



### Advanced User: Do Not Load Screen Rules Versus Disable Screen Rules

The **Do Not Load Screen** rule does not disable that screen. It will simply prevent a given screen from being loaded under certain conditions. If there are mandatory variables on a screen the abstractor cannot load, he/she will not be able to complete the case in data entry (if post processing is turned on). If the variables on a screen should be **disabled** for a given set of logic, use the **Disable Screen** rule.

## Edit Disable Screen/Subscreen Rule

---

The **Disable Screen** rule is created on the **Screen Design** window. Select the screen/subscreen to be disabled and then enter the logic for the rule. Only if the rule executes to true are the variables on that screen disabled; otherwise, no action takes place. (Remember, by default, the variables on a screen are enabled when that screen is loaded.)

Use any logic in the rule including logic for whether a screen is empty (no data entered on the screen) or filled (all mandatory variables have a value). Note, this rule type cannot be used with **Sequential Data Entry**.

**NOTE:**

When using the **Disable Screen** rule, remember that when the disable rule is true, all values entered will be cleared. Therefore, if an abstractor has entered data and then changes an answer that will cause the disable rule to execute to true, all data affected by this rule will be deleted.

## Edit on Entering Case Load Screen Set ? If.. Rule

---

The **On Entering Case Load Screen Set ? IF..** rule is accessed from the **Screen Design** window. The user can select the screen set to load conditionally and then select this rule type from the drop down menu. Enter any logic that will execute when loading the case for data entry. If the rule executes to true, the screen set associated with that rule is the one that is loaded. If the rule does not execute to true, then the first screen set is used by default.

Note that this rule type is typically used when designing a custom **Opening** screen where a screen set is being used to collect a given answer(s) creating the data record (e.g., collecting data for the type of assessment, admission, discharge, revisit); importing a set of records that have a value that is used in the logic to determine which screen set should be used during data entry.

---

## UTILITIES

---

### Edit Medications

---

The **Edit Medications** utility enables a user to maintain the old MedQuest Medications database. This database contains over 30,000 drug names, each of which has a generic name associated with it. Each generic can also be classified into a given code(s), e.g., the drug names "Amcap" and "Amcill" exist in the database and both have the generic name "ampicillin" associated with them. The generic "ampicillin" has a code of "5001" which is in the drug class "Antibiotics." If needed, the generic can be classified into more than one code.

**NOTE:**

The **Edit Medications** utility will allow the user to maintain the old MedQuest Medications database only; no other medications database can be maintained using this utility.

### View Medications

---

When a user selects the **Edit Medications** function, he/she can immediately begin to browse all of the codes, generics, and drugs that are in the Medications database. To view the code, generic, or drug associated with medication, type the search string into the box above each of the lists. This will display the records in the database that match the search string in a list box directly beneath it, e.g., to search for the generics that begin with "asp", type those characters in the box above the generic list.

Once a code, generic, or drug in the list has been identified, use the other browse tools to perform functions such as viewing all of the generics associated with a code, the codes associated with a generic, the drugs associated with a generic, etc. The **View Codes**, **View Generics**, and **View Drug** function

keys will perform these functions when the user has selected the applicable item from the code, generic, or drug list.

## ***Maintain Medication Codes***

---

When building a data entry system, it may be necessary to focus the data collection on a given group of medications. These groups are defined as medication codes. For example, the Medications database contains code 5001 for all antibiotics.

Use the **Maintain Codes Editor** to add or remove generics to or from a given code. When the user accesses this editor, the list of the generics that are linked to the selected code are displayed in the box to the right of the window.

To search for a generic to add to the code, locate the group to which the generic belongs through the **Search by Code** function, or locate the desired generic through the **Search by Generic** function. To search for a generic by code, enter the beginning of a code and select the **Search By Code** function. To search for a generic by generic, enter the beginning of the generic and select the **Search By Generic** function. Each of these functions will display a list of generics that match the search pattern.

To add generics to a code, highlight the generics to be assigned to the code, and select the ">>" function (next to the selected items) to move them into the list of generics for the given code. To remove generics from the list, select the "<<" function. After completing this process, click on the <OK> button to save the changes.



### **Advanced User: Effective Use Of Code Classes**

The classifications of medications can quickly become complex unless a relatively small group of generics is designated to add to a given code. As new drugs constantly become available, it may not be possible to keep up with all of the changes that are occurring in the pharmaceutical and health care industries. This should be recognized as a part of the **Design** process and very clear groupings should be implemented.

Also, the user can explore the possibility of borrowing someone else's medication classifications. For example, in the future, the MedQuest medications database may incorporate many more drug codes from an FDA Medications database that has been made publicly available on the Internet.

Another **Design** consideration is the ability to share these classifications with others. A user may find that by working with others, there is an existing group of medications that is appropriate for their system. MedQuest was designed as an open system that facilitates information sharing among users with the same databases.

## ***Add/Edit/Delete Codes/Generics/Drugs***

---

To change information in the Medications database (i.e., codes, generics and drugs) use the **Add**, **Edit** and **Delete** functions. The user has the capability of adding and deleting any of the codes, generics, and drugs that exist in the database as well as editing the codes. Following is the information that is required and/or optional for each of the entries:

Table	Field	Type
Code	Code (4 digits)	Required
	Description	Required
Generics	Code (4 digits)	Optional
	Name	Required
Drugs	Name	Required
	Generic	Required

The user can enter the required information in the boxes provided when executing the function. To perform the **Delete** and **Edit** functions, select an item from the list.



#### Advanced User: Changes Not Recommended

It is strongly recommended that the user not delete codes, generics, or drugs. In addition, it is recommended that the existing codes not be edited. These records may be in use by others, and any changes made will permanently desynchronize the work from others. It is recommended that the user add new codes (and maintain those codes) as the primary maintenance activity on the Medications database. If it is necessary to add generics and drugs, try to limit the number of changes that are made. If many changes are made and there is no mechanism for sharing this work with others, it may be that when the next version of the medications database is released all of the changes are lost. When deleting a generic name, delete all medications from the DRUGS table linked to that generic.

Because the source database for medications uses only 30 characters for the generic name, some of the combination drugs have shortened compound names to fit into the 30 byte space. It may be tempting to expand them to make them more readable, but the database is currently structured to have unique drug names; that is, there can only be a single drug name. If each compound name had a separate generic record, the integrity of the database, as it exists, would be compromised.

## Edit Units Database

For many clinical variables, the values entered are often represented in different units. MedQuest provides the ability to construct a group of units that may be advantageous in designing the data entry system. These groups of units are accessible in the MedQuest DESIGN menu when building a variable that is a number (with units).

### Add/Delete Groups of Units

A group of units is a convenient way of grouping a set of units together, e.g., it is useful to build a group of units for weight (e.g., pounds, lb, kilograms, kg). When adding a group, enter a unique name for the group and also a description of the group. The group name and group description appear in the **Edit Units** window.

When deleting a group of units, all of the unit information associated with that group is deleted. The information will not be able to be recovered.

### Add/Delete Units to/From a Group

Once the group has been defined, add units to that group. This can be done by selecting the **Add Units In Group** function and entering the unit to add (e.g., "Celsius"). Once this unit is added, it will appear in the list of units associated with the group.

When adding a new unit to the group, that unit can be converted to any other unit in the group. The list of conversions in the group is automatically built and maintained, e.g., if there are two units (pounds and kilograms), MedQuest will provide the conversions of pounds to kilograms and of kilograms to pounds. Note that a default conversion factor of 1.0 is provided unless otherwise specified.

For each of the conversions in the group, enter the linear conversion factor that will be used to convert from one unit to the other. Do this by selecting the conversion in the list provided and then by entering (or changing) the conversion factor in the box provided.

When deleting a unit from a group, all of the conversions in the group involving that unit (and all of their associated conversion factors) are deleted.

## ***Importing Units***

---

MedQuest has a standard set of laboratory units available when selecting the Import function. This function displays a list of industry standard laboratory tests that were developed from information in the ***Stedman's Medical Dictionary*** and from a laboratory database on the Johns Hopkins University Hospital Internet homepage. When selecting a laboratory test in the list, the units that belong to that laboratory test will appear in the list to the right. Note that in some cases the laboratory test will not have any units associated with it.

## **Edit Abbreviations**

---

MedQuest provides an Abbreviations database that contains clinical abbreviations and their acronyms that can be searched during data entry. This database can be maintained by selecting the **Edit Abbreviations** function. When editing abbreviations, it is possible to add and delete abbreviations as well as save changes to an existing abbreviation. Each abbreviation is composed of the abbreviation and the description (e.g., "A & W" is an acronym for Alive and Well). Each abbreviation must have both of these elements entered.

During data entry, access the abbreviation description by typing the abbreviation in the box provided on the **View Abbreviation** window. This function can also be used to browse through all the abbreviations in the Abbreviation database.

## **Edit Keywords Database**

---

MedQuest provides the capability to classify variables into different groups. These groups are defined by keywords. The classification of a variable is a very powerful tool that can be utilized when both building a data entry system and analyzing the data that are produced by that system. Currently, MedQuest uses only the keywords in the **Query** function. The **Query** function enables the user to search for a group of variables by keyword. This enables the user to quickly access variables of the same type and select the ones of interest.

## ***Add/Delete Keywords***

---

Each keyword is defined by a unique keyword name and a unique description, e.g., there may be a keyword name of "UB92" and a description of "UB-92 Variable." When adding a keyword, enter both the name and the description. When first creating a keyword, there are no variables that have been "linked" to it, so a link between a variable and a keyword must be added.

When deleting a keyword, not only is that keyword removed from the dictionary, but any links between that keyword and other variables are also removed. A deleted keyword and its links cannot be recovered.

## *Search for Existing Keywords*

---

Keywords can be searched alphabetically using their description by typing the search characters into the box provided on the screen. The keywords that match this search string will appear in the keyword list. To view variables linked with the keyword, select one of the keywords in the list.

## *Add/Delete Links*

---

A link establishes a relationship that puts a variable in a group of variables associated with the keyword. Variables can be added to the group (add link) or removed from the group (delete link).

To add a variable to the keywords group (add link), first select the keyword from which to add a variable link. Next, retrieve the variable(s) to add to the group. All of the variables in the dictionary can be retrieved. To add a link, select the variable(s) to link and select the **Add Link** function.

To remove a link, first select the keyword from which to remove a link, then select the variable(s) to remove from that keyword group. The **Delete Link** function will remove the link.



### **Advanced User: Take Advantage Of Keywords**

MedQuest currently provides use of the keywords in the **Query** function. This feature may be advantageous during construction of a data entry system, e.g., to create a keyword "Blood Pressure" and link all of the blood pressure related variables ("Hypertension", "Hypotension on admission", "High Blood Pressure", "Systolic Blood Pressure on Discharge") to it. Then when another module is created and a blood pressure variable is needed, use the **Query** function to access the keyword "Blood Pressure" and browse all of those variables. This can also make it easier to ensure consistency across the variables by displaying the help and titles for all of the variables in the group.

The keywords can also be used during analysis or reporting of the data collected, e.g., use SQL to join the "Blood Pressure" linked variables with the data collected for those variables, and then perform an analysis on just the blood pressure related variables.

## *Edit Variable Group*

---

This general tool is used to design the Header Variables, the Data Entry Screen Top, and the Variable Assistant Popup box.

**VARIABLE GROUP EDITOR.** The Variable Group Editor Design window is a general design tool that is used to build a group of variables for different purposes. There are a number of functions that vary in availability and use depending on the requirement of the general tool. These functions are described below.

**VARIABLE GROUP.** The variables that belong to the group being edited are listed in the list box in the editor, e.g., if the variables in the Header are being edited, then those variables will appear in the list.

**ADD VARIABLE.** When adding a variable, if the function requires selecting an existing variable to add to the list, then a list of the screen sets, screens and variables will be provided from which to select a variable. When a variable is selected, it will be added to the list.



When the function requires creating a new variable to add to the list, then the standard variable design window for adding a new variable will be provided (this is described in the section on how to add a new variable to a data entry screen). The same principles apply whether adding a new variable to a data entry screen or adding a new variable to the variable group list.

**EDIT VARIABLE.** This function is only available when the user is allowed to add a new variable in the **Variable Group Editor**, e.g., when adding variables to the **Header** variable group, the user is allowed to create a new variable. This function will open the standard variable design window.

**DELETE VARIABLE.** When deleting a variable, it is removed from the list. That variable can be added to the list by selecting the same variable name using the **Add Variable** function. However, particular system variables cannot be deleted.

**MOVE VARIABLE UP/MOVE VARIABLE DOWN.** Using these functions can reorder the variables in the variable group on the screen.

**EDIT VARIABLE HELP.** The clinical help for the variables that are in the variable group can be edited by selecting this function. This will open the standard variable clinical help editor design window.

## Edit Header Variables

---

When building a module, there is a given set of system variables that is always added to the screen set called the **Header**. This screen set will always exist and the system variables contained in it cannot be deleted. The system variables (e.g., the time of abstraction) in the screen set are not entered by an abstractor on a data entry screen but are created by the MedQuest program.

It may be necessary to add variables to MedQuest that should not appear on a data entry screen, e.g., some of the UB-92 (HCFA claims data) variables such as the medical record number or the admission date that have data available in another source. In this situation, edit the **Header** screen set and add other variables.

Once the design of the **Header** screen set and the data entry system are completed, "download" data into the header using the MedQuest **Import** function. Please refer to the **Import** function description in the MedQuest Manager (SMS) utility.

The ability to add, delete, edit, and move variables in the **Header** screen set group is accomplished with the **Variable Group Editor**. The details of this function were described above.

## Edit Data Entry Screen Top

---

In data entry mode, the top of the screen displays the short title and value entered for up to seven variables. This provides a quick reference to critical variables that may impact data entry on other screens, e.g., view the admission and discharge dates on the top of the screen.

Select the variables that will appear on the top of the data entry screen using the **Edit Data Entry Screen Top** utility. This provides the capability to add, delete, and reorder the variables that appear. Note that only certain types of variables can be selected, since some variable types (i.e., a grid medication variable) will not have sufficient room to display anything meaningful.

A variable may also be deleted from a screen while in the **Design** mode. If this is the last reference to that variable, and the designer chooses to delete that variable from the dictionary, it will also be deleted from the **Data Entry Screen Top** (if it is referenced there).



This function utilizes the **Variable Group Editor** described above.

## Edit Variable Assistant Popup

---

In some instances, a set of variables (and their values) may need to be displayed together for reference purposes. If the screen top is filled with variables and there is a need to provide the user with even more variables to view, add variables to the **Variable Assistant Popup**.

This function utilizes the **Variable Group Editor**. This tool provides the capability to add, delete, and reorder the variables displayed in the popup. The **Variable Assistant Popup** is accessible in data entry by selecting the <VARIABLE ASSISTANT> button that appears to the right of the <PAUSE> button.

## Edit Case Status Window

---

When a data abstractor exits a case, the **Case Status** screen will appear. By default, it contains one variable (i.e., **Actionrs**, a system variable) for marking a case as "Is In Use," "Completed" or "Stopped, Technical Reason." The options for this variable can be expanded (and new variables added to the **Case Status** screen) through the **Edit Case Status Screen** function.

Design the **Case Status** screen just like any other data entry screen in MedQuest. Note that this screen must be designed properly in the WYSIWYG fashion because it will always be displayed in WYSIWYG when exiting a case. When editing the **Actionrs** (Action Reason) system variable, the first two options cannot be modified.

When exiting the case in **Data Entry**, the **Case Status** screen (if requested) will prompt the abstractor to answer the questions provided on the screen. The abstractor will be required to answer any of the questions on the screen that are enabled.

## Edit Opening Screen

---

MedQuest now provides a new type of user interface for creating new records and adding and deleting records by providing the user with the capability to design a custom **Opening** screen. It provides a different approach to managing the **Case Selection** window. The current screen (known as the **General Opening** screen) only allows the user to add his/her own unique identifier (and search by that identifier), and is useful when importing his/her own records or using similar processes. The new interface provides a method for adding a unique record identifier and also searching for an existing identifier.

Designing a custom **Opening** screen for a data entry system provides the look and feel of a customized module. When distributing this "customized module" along with the MedQuest Data Entry software, the user can distribute his/her module as a stand-alone data abstraction instrument.

## Prepare the Module for an Opening Screen

---

**DEVELOP A LIST OF HIERARCHICAL OBJECTS.** Determine the hierarchical order of the data, e.g., a hospital has departments, departments have patients, and patients have data records.

**ADD UNIQUE IDENTIFIERS TO THE HEADER.** Select the Edit Header Variables option from the UTILITIES menu item. Create a unique field for each of the objects to be included in the hierarchy. These unique field names will link the hierarchical (parent) objects to the data records, e.g., add HOSPID and PATID to the header to link the Hospital and Patient objects to the data record. If necessary, review

the Edit/Download Header Variables section listed above. When adding these variables, they must be specified as Numeric (no units), whole digits equal to 10, and decimal digits equal to zero.

**CREATE SCREEN SETS.** Create a screen set for each hierarchical object (in this example, one for Hospital and one for Patients) and the screen set to be used when a data record is added (Add Data Record).

**CREATE VARIABLES.** For each of the screen sets, add screens and then add the variables that will collect the desired information, e.g., the **Hospital** screen set might contain variables for Name of Hospital, Location, etc.; the **Patient** screen set might include demographic information about the patient (Name of Patient, Medical Record Number, etc.); and the **Add Data Record** screen set could include data elements related to a specific hospital event. Each event could have a separate record created. This screen set will contain the bulk of the variables for collecting information about the patients.

**ARRANGE THE SCREEN SETS.** Arrange the screen sets according to the hierarchical layout. For instance, the **Hospital** screen set would be first, followed by the **Patient** screen set, and then the **Add Data Record** screen set. Although not necessary, it is advantageous to place the screen sets in this order for design and review purposes. For design, the **Opening** screen will only work with the first 250 variables in the dictionary. The order of the variables in the dictionary is established by screen set order with header variables included first. For review, the design of an opening screen will be easier to analyze when the screen sets are ordered according to the hierarchical design.

## *Edit the Opening Screen*

---

While in **Design** mode, access the Edit Opening Screen option from the UTILITIES menu item. The **Design Opening Screen** will be displayed.

The **Design Opening Screen** window displays a list of objects (the hierarchical list: Hospital, Patient, and Add Data Record) in the *Objects on Opening Screen* text box. The DATARECORD is the default object that appears when first designing the **Opening** screen. The DATARECORD is always the last object in the hierarchy and is where the data for a module are collected. The name DATARECORD cannot be changed, and the DATARECORD object cannot be deleted.

To add an object to the hierarchy, select the object to add above (this would be the DATARECORD), and click on the <ADD (ABOVE)> button. The hierarchy object called NEW is added to the Objects on Opening Screen text box. To delete an object, highlight the object and click on the <DELETE> button. A message will appear asking the user to confirm the deletion. To reset the **Opening** screen back to the default settings (no custom screen), click on the <DEFAULT (NO CUSTOM SCREEN)> button. A message will appear asking the user to confirm the request. When exiting the **Design Opening Screen** window, click on the <OK> button to save any changes or the <CANCEL> button to discard any changes.

Listed below are the object properties required for completing a custom **Opening** screen.

## **Object Properties To Be Entered For The Opening Screen**

---

**OBJECT NAME.** Enter the name of the object (i.e., Hospital) that will be used to display the object in the data entry browser. Each object name must be unique.

**TABLE NAME.** For each object above the DATARECORD object, enter a table name (i.e., Hospital) in the *Table name (where object data saved)* list box. This table will hold the data entered for the HOSPITAL object. It will be created when the datastore database is rebuilt.

**NOTE:**

Note that the default table name for the DATARECORD object is the three letter acronym of the module that is being created.

**SCREEN SET.** Create a screen set for each of the objects in the hierarchy. Select the name of a screen set from the drop down list displayed next to the *Screen Set Name ...* list box. This is the screen set that will be loaded for data entry when an object record is added on the **Opening** screen, e.g., select the **Patient** screen set (containing the First Name, Last Name, and SSN variables) for the PATIENT object. When the PATIENT object record is added, the **Patient** screen set containing the Patient specific variables will be loaded.

To finish defining an object, it is useful to select a screen set since the variables added to the other fields will be called from the selected screen set. The DATARECORD object is the exception. The user may select variables from any or all of the available screen sets for the DATARECORD object. If a screen set is not selected, then when the object record is created on the **Opening** screen, only an internal unique identifier is added to the record. This is useful for designating a parent object to be used only as a counter.

**NOTE:**

A screen set may only be used once when creating an object for the **Opening** screen.

If the user changes to another screen set and the variables listed in the object fields are not included in that screen set, the variables will be deleted from the fields. (This does not apply to the DATARECORD object.) This is to ensure that the user does not include variables that are not applicable to the selected object.

**UNIQUE FIELD.** To add the unique field (created when the unique identifiers were added to the **Header**), click on the <+> button to display a list of the variables that were added to the **Header**. Select the variable that was created specifically for the object that is being created (i.e., Hospital for the HOSPITAL object). Each variable created for an object may only be used once.

**UNIQUE RECORD.** This field is only enabled for the DATARECORD object. The user can select a field(s) (variable) to uniquely identify a data record that is added during the data entry process. Click on the <+> button to display a list of variables from which to choose. To delete a field click on the <-> button. Use the <UP> and <DN> buttons to move the variables up or down in the list.

**COPY FIELDS.** When an object is added during data entry, it may be advantageous to copy specific fields from that object to the data record. For example, to copy the Patient's Name to each data record, highlight the PATIENT object from the *Objects on Opening Screen* text box and click on the <+> button to display a list of the variables (associated with that object) from which the user can choose the **Patient's name**. To delete a field click on the <-> button. Use the <UP> and <DN> buttons to move the variables up or down in the list.

**DISPLAY FIELDS.** These are the fields that will be displayed on the **Opening** Screen for each data record that is added, i.e., the Patient's Name, SSN, Admission Date, etc. The variables are displayed from left to right and are separated by commas. Click on the <+> button to display a list of variables from which to choose (only the unique field and the variables associated with this object will be available to add). To delete a field click on the <-> button. Use the <UP> and <DN> buttons to move the variables up or down in the list.

**SEARCH FIELDS.** These are the fields that will be displayed on the **Opening** screen from which the user can search for an object. Click on the <+> button to display a list of variables from which to choose (only the unique field and the variables associated with this object will be available to add). To delete a

field click on the <-> button. Use the <UP> and <DN> buttons to move the variables up or down in the list.

**ORDER FIELDS.** These fields will be displayed on the **Opening** screen in the order specified by the user. The user may designate ascending or descending order by selecting the corresponding radio button. Click on the <+> button to display a list of variables from which to choose (only the unique field and the variables associated with this object will be available to add). To delete a field click on the <-> button. Use the <UP> and <DN> buttons to move the variables up or down in the list.

## Options That Apply to All Objects

---

**DISPLAY POST PROCESSOR WARNINGS.** The user may select to display the post processor warnings for each object when data entry is complete for the specified screen set.

**DISPLAY CASE STATUS SCREEN.** For each object, the user may select to display the **Case Status** screen when data entry has been completed for the specified screen set.

**UPDATE ALL DATA RECORDS (NOT APPLICABLE TO DATARECORD).** The user may select to copy a change made to an object in the hierarchy to all associated, or linked, data records, e.g., a name change was made to a PATIENT record and the name variable is also on the DATARECORD as a read only variable. All of the associated data records for that patient can be updated. The user may elect to update all data records for object changes with or without a warning or choose not to update.

## Options That Apply to Data Record Only

---

**DISPLAY SCREEN SETS.** This option allows the abstractor the capability to choose from more than one screen set during data entry. These are the screen sets that are associated with the DATARECORD (data entry) not the screen sets associated with the Custom objects. If the user has invoked the "On Entering Case Load Screen Set ? If.." rule, then this option would not be selected.

**UPDATE DATA RECORD ON EDIT.** When editing a data object, the data for a parent object (i.e., PATIENT) may have changed since the data record was last edited. The user can select to copy the change into the data record. The user may select to update all data records for object changes with or without a warning or choose not to update. This feature enables the user to maintain data integrity during data entry.

## Rebuilding Object Tables

---

When exiting the **Design Opening Screen** window, a rebuild of the data store database will take place to update the tables with the most recent design changes. The tables associated with the parent objects can only be built per the specifications entered during design.

### NOTE:

If the specifications for the parent objects are changed after entering data, e.g., if the unique field for HOSPITAL or PATIENT is changed, the links connecting the parent tables and the data will be lost.

---

# PREFERENCES

---

Preferences provides the user with choices that apply to the data entry system as a whole. With most of the choices, the options can be switched on (checked) or off (not checked) by selecting that choice from the OPTIONS menu. In some cases, the user will be required to enter additional information.

## General Options

---

There are a variety of options the user can set. These include:

### *Variable Enable/Disable Features*

---

The user has a choice to turn on or off certain enable/disable features, including:

**WARN USER BEFORE DISABLING (AND CLEARING DISABLED) VARIABLES.** If this option is selected, when the data abstractor changes the value of a "parent" variable that has enabled or disabled the data entry value of subsequent related variables, the data will be cleared from these related variables. A warning message gives the abstractor the option of disabling and clearing any data entered in these related variables or of leaving the answer to the "parent" variable as is. Note that this option is not available for **Sequential Data Entry**.

**CLEAR VARIABLES (AFTER ACTION VARIABLE) IF EXIT CASE RULE IS TRUE.** If this option is selected, when the data abstractor invokes the **Exit Case Rule** as true, the values of all subsequent variables will be cleared. A warning message gives the abstractor the option of exiting and clearing the values of all subsequent variables or of simply exiting the case.

**CLEAR GRID VARIABLES IF DISABLED.** If the user selects this option, then whenever a grid variable is disabled, all of its records will be deleted as well.

### *Data Entry Alternative Screens*

---

The user has a choice to turn on or off certain screens that appear during data entry. These optional screens are:

**CASE STATUS SCREEN.** If the user selects this option, the **Case Status** screen appears after **Data Entry** is complete. This screen requires the data abstractor to, at a minimum, mark the case as still in use, complete, or stopped, technical reason. In addition, the user can add questions that will be asked on this screen.

**POST PROCESS AFTER CASE.** If the user selects this option, then any post processor warnings will be displayed after the data abstractor completes **Data Entry** and exits the case.

**POST PROCESS AFTER SCREEN.** If the user selects this option, then any post processor warnings will be displayed after the data abstractor has completed the data entry on a screen and tries to switch to another screen.

**POST PROCESS AFTER GRID.** If the user selects this option, then any post processor warnings will be displayed after the data abstractor has completed the data entry on a grid and clicks on the <CLOSE> button.

## Unique Case Identifier Properties

---

This feature allows the user to specify the Case ID length, from 12 to 250, and the Reabstraction Case ID length, from 1 to 9. The default value for Case ID is 40 and the Reabstraction Case ID is 1.

### NOTE:

If the user has specific fields for a custom **Opening** screen data record to uniquely identify that record, the Case ID length will increase as necessary.

## UTD Information

---

This feature lets the user choose the UTD character. The default value is "X." In data entry, this character is selected when the user presses the <SHIFT> and <?> keys.

## Colors

---

The user has the option of specifying the color that certain conditions of the variables will invoke. These include: Read Only Variables, Mandatory Variables, Empty Variables, and Filled Variables. The Read Only and Mandatory Variable colors take precedence over Empty and Filled Variable colors.

### NOTE:

The user also has the capability to make a given color not applicable. This is useful to ensure that the colors (e.g., empty and mandatory settings) do not conflict with each other. By default, if a color is not specified, the Not Applicable option is checked.

## Miscellaneous 1

---

There is a variety of screen display options that the user can set. These include:

**NUMBER VARIABLES.** If the user selects this option, the variables will be numbered (in their natural order) on the screen during design and data entry. They will also be numbered in the **Dictionary Survey Report**.

**SHOW FIELDNAMES.** If this option is selected, the 32-character variable name will appear on the screen during both design and data entry as well as in the reports.

**SEQUENTIAL DATA ENTRY.** Selecting this option will allow the data abstractor to enter data sequentially. As the user develops his/her data entry system by adding screens and variables, when a case is entered, only the first variable on the first screen will be enabled. When this variable is answered the next one will be enabled, and so on.

If sequential data entry is selected, there cannot be any indented variables, any screen rules, or any enable or disable rules. However, skip to rules can be used.

If a data entry system is created and then changed to a sequential data entry system, there will be a warning given to remove all rules, excluding skip to rules, and all indentations. The variables will have to be realigned by screen using the **Align Variables** function (if the indentations are also represented graphically on the screen).



**NOTE:**

Sequential data entry is not available for the **Case Status** screen. However, the user can use enable/disable rules for this screen.

**DEBUG ON.** Selecting this option invokes a **Debug** window in data entry. The **Debug** window lets the user view the rules that were added to the data entry system. Depending on the types of rules added, they would be displayed in the window as they are invoked. For instance, if an "Exit Case If" rule was used in the data entry system, when a case is opened and the first screen is loaded, the equation for this rule will appear in the **Debug** window. The rule will be equal to false since the variable containing this rule has yet to be answered.

By turning on this option, the user will be able to view the rules that were entered and observe the execution of those rules. This will help the user to review the logic and debug it, if necessary. It is suggested that the logic (data entry rules) be reviewed frequently. Invoking these rules periodically by entering a case will help to isolate any conflicting logic, especially if the data entry system is large or complex.

**COMPILE DICTIONARY.** Selecting this option will "compile" the dictionary to speed up its loading during start up. It adds additional fields to the data dictionary, and each time the user exits design, the dictionary is "recompiled" by updating these fields.

**USE CHECK BOXES (NOT RADIO BUTTONS) FOR OPTION (PICK ONE).** Selecting this option will convert all Option (Pick One) variables containing radio buttons to switch to check boxes. This feature will allow for tabbing between variables.

**TAB COLORS.** Turn on tab colors for filled or partially filled screens. For example, the user can select to color a tab blue if any data is entered on the screen or color a tab green if the screen is filled. If this option has been selected, all of the enable, disable, skip to and disable screen rules will execute for all of the screens. Note that using this option will slow down the processing. This option is not available for Sequential Data Entry.

**NOTE:**

Tab colors are displayed when a case is loaded or when a screen is exited to update its color. This means that a tab's color will change only when the screen has been exited.

**ALLOW VGA DATA ENTRY ON SCREENS DESIGNED AT HIGHER RESOLUTION.** Selecting this option allows users to design modules using a higher resolution (800x600 pixels). During design and data entry, for users who have lower resolutions (640x480 pixels), scroll bars will be displayed to enable them to access all the variables on a screen.

**UTD CHOICE ON EACH ITEM IN OPTION (PICK ONE OR MORE).** Selecting this option will give the abstractor the capability to tag each item in an Option (Pick One or More) variable list as UTD. In data entry, the abstractor can right mouse click on the desired option to mark it as UTD. A low highlighted check will be displayed to indicate that the option has been marked as UTD.

**NOTE:**

This feature is not compatible with the MedQuest Quality (IQC) program.

**SWITCH TO LISTBOX WHEN NUMBER OF ITEMS IN OPTION (PICK ONE) OR OPTION (PICK ONE OR MORE) IS GREATER THAN ?.** The user has the capability to designate the number of options to list for a variable before it converts to a list box. It is recommended that this not exceed more than 25 options. The default value for this feature is 15 options. The maximum allowable options for a list are 99.

**DICTIONARY VERSION.** The user has the ability to designate the version number for a dictionary. The default value for a new dictionary (data entry system) is 10 and is limited to 8 characters (i.e., CCP1.00).

Version control of a dictionary is very important especially when sharing a dictionary or using it for data abstraction. It is necessary to know who has control and if any changes are being made. Once data abstraction has begun on a dictionary and then changes to that dictionary are made, the reliability of the data and any analyses that have been done can be compromised.

## ***Miscellaneous 2***

---

There is a variety of screen display options that the user can set. These include:

**DO NOT INCLUDE GENERAL DATA ENTRY GUIDELINES MENU OPTION.** The module developer has the option of creating a help file describing the general guidelines used for data entry. If the developer does not want or need to create this help file, it can be disabled. The user will not be able to access the General Data Entry Guidelines option from the HELP menu item.

**DO NOT LOAD MODULE ACRONYM TO VARIABLE NAME WHEN ADDING A VARIABLE.** The user has the capability of not including the first three characters of the module acronym in a variable's name during the design process. This is useful if a module is created whose first character is a number instead of a letter. Due to the restrictions in the software, the first character of a variable's name must begin with a letter. Therefore, to avoid having to erase or change the first character of every variable's name that is added to the module, select this option.

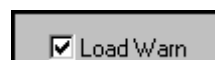
**# OF FIELDS PER TABLE (MAX 250).** The default number of fields per datastore table is 250. In certain situations the user might find it useful to reduce this number, e.g., when using the ODBC feature, the limit for number of fields is approximately 100. By allowing only 100 fields per datastore table in a module, it will be compatible with the ODBC function.

**IQC DATABASE NAME.** The user has the capability to specify the file name of the IQC database. This file name is used to create the IQC database when starting the MedQuest Quality utility for a new module, or it is used to lookup an existing IQC database for an existing module. Up to eight alphanumeric characters can be entered for the database name. The default database name is IQC to maintain the integrity of existing modules.

**IN IQC WARN USER ON LOAD IF ORIGINAL OR REAB DATA HAVE CHANGED.** The user has the capability to turn on a warning message in the MedQuest Quality Program when loading a comparison. This feature will automatically detect if there were any changes to the data for the comparison set being loaded since a comparison was run. This option checks the abstraction date, abstractor ID, and review time in the IQC database and in both the Case 1 and Case 2 databases to determine if they have changed.

### **NOTE:**

This option should not be used when saving the adjudicated values to the Case 1 database since the Case 1 database is changed every time adjudication is performed and the warning will constantly be displayed.



While in the MedQuest Quality Program, from the **IQC Program** screen the user has the option to deselect this feature. Uncheck the *Load Warn* check box to disable this option. If this option is unchecked, it will only be disabled for the current MedQuest Quality session.



## NOTE:

If this feature is not selected in MedQuest, the *Load Warn* check box will not be displayed on the **IQC Program** screen.

**IN IQC SAVE ADJUDICATED VALUE TO THE CASE 1 (ORIGINAL) DATABASE.** This feature will automatically save an adjudicated value to the module database for Case 1 during the adjudication process in the MedQuest Quality utility. Normally, the adjudicated data are not saved to the module (i.e., original) database. Therefore, it was recommended that the MedQuest data entry be used to perform this update (if required). However, the MedQuest tool gives the user the option to save the adjudicated value to the module database for Case 1 when exiting adjudication.

Checking the *In IQC Save Adjudicated Value to the Case 1 (Original) Database when Exiting Adjudication* check box will display the *Only Save for ReabID=0 or Blank (No Reabstraction Data)* check box. Since the MedQuest Quality gives the user several options for comparing cases, it is recommended that the check box *for Only Save for ReabID=0 or Blank (No Reabstraction Data)* also be selected so that only original data are saved to the module database.

When the adjudicated value is saved, the abstraction time is incremented by one second and the current date is saved to the database. This will verify that the database has been updated.



While in the MedQuest Quality Program, from the **IQC Program** screen the user has the option to deselect this feature. Uncheck the *Adjud. Save* check box to disable this option for the current session.

## NOTE:

If this feature is not selected in MedQuest, the *Adjud. Save* check box will not be displayed on the **IQC Program** screen.

**IN IQC ARCHIVE THE CASE 1 (ORIGINAL) DATA.** This feature will automatically save the original data for Case 1 to an archive database. Checking the *In IQC Archive the Case 1 (Original) Data when First Loading a Comparison Set* check box will display the *Only Archive for ReabID=0 or Blank (No Reabstraction Data)*. Since the MedQuest Quality gives the user several options for comparing cases, it is recommended that the check box *for Only Archive for ReabID=0 or Blank (No Reabstraction Data)* also be selected so that only original data are saved to the module database. If an archive database does not exist for a module when this feature is selected, the MedQuest Quality will create it the first time the module is loaded into the program.

## Medications

---

The user has a choice of selecting between two different medication databases to use during module development.

**USE OLD MEDQUEST MEDICATIONS DATABASE.** The old MedQuest database was released with MedQuest prior to version 4.10. This medications database will no longer be maintained. However, there is the option to continue using it as a primary source of medication names for verification during design and data entry. The **Edit Old MedQuest Medications Database** utility is provided with the MedQuest software so that the user can continue to maintain the database if this option is selected.

**USE MULTUM MEDISOURCE™ LEXICON MEDICATIONS DATABASE.** This database will be included with the release of MedQuest 4.10 and subsequent releases. It is a commercial database in MS Access available on the Internet ([www.multum.com/technical.htm](http://www.multum.com/technical.htm) - requires on-line registration) free of charge that is updated weekly with the latest data from pharmaceutical manufacturers. Due to its size (35MB),

some of the tables containing information extraneous to the process of collecting clinical data have been removed. If there is a need for the types of data that are currently not included, these can be added in a future release. At this time, categories are provided by the database developer and cannot be modified by MedQuest users.

**NOTE:**

This tool uses the MediSource™ Lexicon database licensed by Multum Information Services, Inc. ("Multum"). Modifications have been made to this database and there is no warranty on these changes. Please see a copy of the MediSource™ Lexicon license agreement from Multum.

**INCLUDE OLD MEDQUEST MEDICATIONS DATABASE.** Selecting this option will include the old MedQuest Medications database during module development and data entry. It can be used to create custom medication lists. (See the Add a Grid Variable under the Design Functions section.) It can also be utilized as an additional source of medication names during data entry. However, the classifications that are included in this database cannot be accessed during module design or data entry when this database is used as an adjunct.

**USE ACRONYMS TABLE (USER MAINTAINED ACRONYMS TABLE IN MMEDS.MDB).** Selecting this option will include the user defined acronym table.

---

## EXIT DESIGN

---

When exiting the MedQuest **Design** function, if mandatory changes to the dictionary have been included that require the module datastore database to be rebuilt (e.g., a new variable was added), then the system will notify the user that the Rebuild function will be performed. The user will not be able to exit the system without rebuilding. This is to ensure that the dictionary (XXXDICT.MDB) and the datastore databases (XXX.MDB) are always the same version.

---

# DATA ENTRY

---

---

## DATA ENTRY - OPENING SCREEN

---

MedQuest lets the designer customize the opening screen or use the default general opening screen. Each of these opening screens can help the user select or create a record for data entry. The features available in the **General Opening (Default)** screen and the **Custom Opening (User Designed)** screen are described below.

---

### Data Entry - General Opening (Default) Screen

---

If the custom **Opening** screen is not used, it will be necessary to create a unique case (record) identifier for each case that is added. The default **General Opening** screen is most useful when importing cases from a sample before data entry.

When the MedQuest Data Entry option is selected, there are several functions that can be performed. These functions can be activated from command buttons and menu options on the screen.

**DATA ENTRY.** To perform data entry, access a selected case by typing the unique case identifier into the *Case ID:* box. The case number that matches closest to the pattern that is being typed (as well as up to 200 cases following in alphanumeric order) will be displayed in a list directly below the entry. Enter an existing case number in the search box, select an existing case number from the case number display box, or enter the number of a new case to begin data entry.

Begin data entry for a selected case by clicking on the <OK> button. If the module has more than one screen set listed, select a screen set before clicking on the <OK> button.



#### **Advanced User: Enter Data For A System With More Than One Screen Set**

A user can enter data for more than one screen set in the same case. Select one screen set, begin data entry, exit the case and select another screen set, and then enter data for the new screen set. Before designing a module, it is advisable to think about a strategy for the screen sets and how they will be used during the data entry process. MedQuest will save all of the data that have been entered. The last screen set that was used will be the one identified as the current screen set when exiting and reentering the MedQuest system.

#### **NOTE:**

It is now possible to use the "On Entering Case, Load Screenset If.." rule for a screen set to determine which screen set to load when adding or editing a data record. The user then is not required to select a screen set.

**DATA ENTRY REABSTRACTION.** Abstracting data for a case more than once for quality control purposes (e.g., to compare the answers entered by two different data abstractors), can be done by entering the existing case number in the *Case ID:* box and then assigning a unique single character into the *Reab ID:* box. When beginning data entry, a reabstraction case will be created that contains all of the required system data and all the downloaded (header) data from the original case. None of the data that was entered in the original case will be copied.

If the *Reab ID*: box is left blank or a 0 (zero) is entered, the case is considered to be an original case and not a reabstraction.



#### Advanced User: Select A Reabstraction ID

Plan the use of the reabstraction identifier carefully. Since the MedQuest Quality (IQC) software program is used to compare two cases, and was designed to work with reabstracted cases, it is important to understand the functionality of that program before taking advantage of the reabstraction identifier. Typically, characters A-Z or 1-9 have been used to identify reabstraction cases.



#### Advanced User: What Is Copied To A Reabstracted Case

When creating a reabstraction case, all of the data in the **Header Screen** set is copied to the reabstraction case. This includes the pertinent system information (e.g., the case identifier, the reabstraction identifier, etc.). Information that is unique to the reabstraction (e.g., abstraction time or user identifier) is not copied. However, any fields that the module designer has included in the header (e.g., claims data variables) will be copied.

#### EXAMPLE:

If a user has added some claims variables and breast cancer registry data to the **Header Screen** set during the design, use the MedQuest Manager (SMS) utility to import those data into the datastore database. When creating an original or a reabstraction case, all of the data imported from the claims data and breast cancer registry data will be copied to the new case.

## Data Entry - Custom Opening Screen

---

A custom **Opening** screen presents the user with a browser-like interface for selecting, adding, and editing objects. The objects are displayed as a hierarchy. All objects in the hierarchy can be linked by means of unique identifiers. The highest level in hierarchy is always labeled "TOP." This cannot be deleted or renamed. All user-defined screens are placed below this level.

When the **Opening** screen is loaded, up to 200 records that occur one level down from the TOP are displayed. For example, if a hierarchy of objects contains three levels, Hospitals, Patients, and Data Records in that order, the first 200 Hospital records, will be displayed.

### Browse the Objects

---

Since the user can add a hierarchy of objects (see the Edit Opening Screen section), the browser displays both the hierarchy and the objects (e.g., Hospital contains Patients which contains data records). When selecting an object in the browser, all the objects beneath it are displayed. For example, if the user selects TOP, all objects at the highest level (for example, Hospitals) will be displayed. If the object is a data record, no other objects will appear beneath it since this is the lowest level.

#### NOTE:

When the *Display Reabstractions* check box is selected, a reabstraction case will be indented beneath the original case when the parent object (Patient) is selected.

By using the scroll bars and selectively highlighting an object to view, the user can select an object to add, edit or delete. As different objects are selected, the search, add, edit, and delete command buttons adjust to be context sensitive to the type of object selected.

Select Object Below to work with	
TOP	
└HOSPITAL=0000000001	
└PATIENT=Casper Z Ghost,123456789ABC	
└DATARECORD=Johns Hopkins,123456789ABC,Casper Z Ghost	

For example, selecting the data record object enables the user to add a reabstraction (add button displays <ADD REABSTRACTION>), edit the current data record (edit button displays <EDIT DATARECORD>), and delete the current record (delete button displays <DELETE DATARECORD>). Since this is the lowest object in the hierarchy, the **Search** mode for a data record is disabled.

Add REABSTRACTION		Reab ID:	<input type="text"/>	<input type="checkbox"/> Display Reabstractions
Edit DATARECORD		Delete DATARECORD		

The fields that can be displayed for each object are designated during the design process. Each of the fields' values (e.g., Patient displays the patient's name ) are displayed in the order in which they were listed during the design. Each item in the list starts with the object name. For reabstractions, the unique reabstraction identifier is displayed.

## Search for an Object

Once the **Opening** screen has been loaded, up to 200 records that occur one level below the TOP (highest level on the browser) are displayed. For example, if a hierarchy of objects contains three levels, Hospitals, Patients, and Data Records in that order, the first 200 Hospital records will be displayed.

The <SEARCH> button is context sensitive to the type of object selected. When the parent object is selected (i.e., a patient) the search button will allow the user to search for the children (i.e., a data record). A data record will disable the search button since it is the lowest item in the hierarchy, unless the data record is the only level (beneath TOP) available.

Search for DATARECORD	With	Name of Hospital
<input type="text"/>		Medical record number
		Patient Name

To perform a search for an object, select the object under which to search, select the variable to search by, enter a search string (or leave it blank, in which case all records are searched for) and click on the <SEARCH> button. All of the objects (up to the first 200) meeting the search criteria will be displayed beneath the selected object.

## Add an Object

An object may be added beneath the selected parent object (e.g., add a Hospital object beneath the TOP). If the design requires the user to enter additional information (e.g., a hospital name, address, city, state, etc.), the data entry screen(s) for these variables will appear. Otherwise, a numeric record will simply be added as an identifier. When data entry is completed, the user will be warned of any errors (if post processing is specified in design) and a **Case Status** screen will appear (if specified during design).

When adding a data record, the user may be required to answer a set of questions prior to performing data entry. This is often done so that a specific set of questions can be displayed for data entry based on the answers provided when the data record was added. At this time, any fields that were designated from

a parent object for copying to the data record are added. For example, the hospital name and patient name may be copied to the data record from the Hospital object and the Patient object.

**NOTE:**

When adding a reabstraction record, the user must enter a Reabstraction ID in the *Reab ID* text box (a single character (i.e., A) that designates the unique Reabstraction ID).

## ***Edit an Object***

---

When editing an object above the data record and/or reabstraction record in the hierarchy, the questions (variables) that appeared when the object was first added will be displayed. If the user makes any changes to these questions and the developer has specified that these changes be copied to all the related data records (e.g., all data records for a given patient) then the next time the related records are edited, they will be updated with the new data.

Also, if specified during design, a warning message may be displayed to the user before the new data are copied. If the warning is displayed, the user has the option to copy the new data or ignore it. If no warning is given, the new data will be copied automatically.

Selecting the <EDIT DATARECORD> button takes the user into data entry so that all of the required information for that record can be entered. Usually the data record has a large number of questions that the designer has specified. Note that these questions can (and almost always are) different than the questions that appear when adding a data record object. This is because when ADDING a data record, the user is usually being asked to answer a relatively small set of questions that are used to determine which screen set (or set of questions) should be used when completing the record.

When editing a data record that has parent objects, if there are any fields that the designer wanted to have copied into the data record from the parent objects, they will be copied, e.g., the patient's first and last name may be copied into the data record. A warning will appear that these copies are being made (if the designer selects this).

## ***Delete an Object***

---

When deleting an object, that object (and all of the children objects beneath it) will be deleted, e.g., if the user chooses to delete data for a given Hospital object, all of the Patient objects and their data records (and reabstractions) will also be deleted.

## ***Select a Screen Set***

---

Screen sets will only be displayed if the developer has designed a module from which the user can select a screen set to perform data entry. If there are **any On Entering Case Load Screen Set ? IF..** rules, the user will not be required to select a screen set. If a list of screen sets does not appear, this means that the developer has selected the screen set that will be loaded for data entry.

When editing a data record and different sets of questions appear, it is the intention of the developer to have the user answer different questions for different conditions, e.g., in a nursing home there might be a different set of questions for admission than for discharge.

## **Data Entry General Features**

---

The following features are available on both the **General** and **Custom Opening** screens.

**DATA ENTRY (VIEW ONLY).** This option allows a selected case from the *Case to Load* list box to only be viewed. The user will not be able to enter or change any data in the selected case. While reviewing the case, click on the <NOTES> button to type any comments into the *Notes* box.

**NOTE:**

The User ID on the case cannot be changed and the abstraction time will not accumulate.

**SELECT MODULE.** The user can select another module for data entry by choosing this button.

**DISTRIBUTE DATA.** Selecting this option (if available) from the FILE menu item will display the *Distribution Data File* popup box. Currently, the only available option is Copy to Diskette. (The Distribute Electronically option will be available in a future release.) By default, the Copy to Diskette option is enabled. Select the directory location from the Copy to Diskette Options and click on the <OK> button to copy the datastore database to diskette.

**NOTE:**

To e-mail the data for a specific project (i.e., CCP), attach the datastore dictionary file, CCP.MDB.

**VIEW/PRINT REPORTS.** Several reports can be produced from the dictionary: a list of variables; a detailed dictionary report; a survey that can be used to collect the data for a module on paper; a comparison report that produces a list of discrepancies between two different versions of a given module; a data analysis by variable report; a report listing the data entry rules; and a clinical help report. This report function also allows the user to compile a Microsoft help file (.HLP) that organizes the help by screen set, screen, and variable. In addition to reports from the dictionary, this function enables the user to produce a case summary report of the data that have been entered as well as an abstractor summary report. The user also has the capability to design reports based on his/her specific needs and ISO reports that display all the variables that had ISO attributes assigned to them during the design process.

**EXAMPLE:**

To provide a paper data entry form to mail to 10 data abstractors at a hospital site, select the **Dictionary Survey** report after building a module. This produces a paper data collection instrument that matches the data screens to be completed using the MedQuest **Data Entry** function.

**QUERY VARIABLE.** This function enables the user to browse through the variables and their clinical help for all of the modules listed in a project database. There are three search mechanisms available: the user can search for a variable by module when the screen and variable are known; through all of the variables in a selected module by using a word or phrase that might be included in the variable's title; and using the keywords that have been associated with a variable in a given module's dictionary. These keywords can be added during module design to create a more robust dictionary variable classification scheme.

**IMPORT.** The **Import** function is accessible through MedQuest. Selecting this option will take the user directly into the MedQuest Manager's (SMS) Import function. See the MedQuest Manager's (SMS) Reference manual for a complete discussion of the **Import** function.

**EXPORT.** The **Export** function is accessible through MedQuest. Selecting this option will take the user directly into the MedQuest Manager's (SMS) Export function. See the MedQuest Manager's (SMS) Reference manual for a complete discussion of the **Export** function.

**LOAD CASES.** The user has the capability of loading data records into the currently selected module from another datastore database for the same module, e.g., the user can access the **Load** function to add data records that have been collected at other sites and have been sent via E-mail or on diskette.



From the **Cases** screen, click on the <LOAD> button or select the Load option from the OPTIONS menu item. The **Load Cases** screen will be displayed.

The user can access the datastore database (XXX.MDB) to load from by entering the path/file name into the *Load Cases From* text box or by clicking on the <FILE FOLDER> icon and selecting the path/file name. Clicking on the <FROM ARCHIVE> button will access the archive database for the selected module; this assumes that an archive file has already been created. After the path/file name has been entered, the user can load the cases into the Original datastore database or the Reabstraction datastore database by selecting the respective radio button from the *Load Into* section. The path/file names for the load into databases will automatically default to the current directory location.

There are additional options available when loading cases. The user can choose to add the records to the archive file as well, by checking the Add Records to Archive When Loaded option. If the archive file has not yet been created, MedQuest will prompt the user to create it and add the cases. The option Do Not Add Reabstractions to Originals or Originals to Reabstractions will prevent reabstraction cases from being copied into the original datastore database and original cases from being copied into the reabstraction datastore database.

The user also has the option to either update or not update duplicate records. Choose to load duplicates with or without a warning or not to load duplicates at all.

The final option available when loading cases is the ability to load cases based on a specific rule (logic). By default, the *All* radio button is enabled from the Records to Load section. If the user wants to load cases based on logic such as a date range edit, he/she can select the *Select Records Where* radio button and then click on the <EDIT LOGIC> button. The **Edit Rules** screen will be displayed. From here, the user can enter in the logic to load cases that are based on a specific rule, i.e., load case if the admission date is less than or equal to a specified date. See the Edit Rules (Rule Editor) segment in the Design section of the MedQuest Reference manual for a complete discussion of how to add/edit rules.

After completing the information for loading cases, click on the <OK> button to begin loading the cases or the <CANCEL> button to cancel loading the cases. When all of the cases are finished loading, a popup message box will display the number of records added to the destination file and the number of duplicate records detected (duplicate records are not added).

**ARCHIVE CASES.** The user has the capability of archiving the data records for the currently selected module. From the **Cases** screen, click on the <ARCHIVE> button or select the Archive option from the OPTIONS menu item. The **Archive Cases** screen will be displayed.

The path/file names for the *Archive From* databases will automatically default to the current directory locations for the module. The *Archive To* path/file name will also default to the current directory and the archive datastore database name (XXX-ARC.MDB). The user cannot change the *Archive To* path/file name location.

Another option available for archiving cases is the ability to archive cases based on a specific rule (logic). By default, the *All* radio button is enabled from the *Records to Archive* section. If the user wants to archive cases based on logic (for example, a date range edit), he/she can select the *Select Records Where* radio button and then click on the <EDIT LOGIC> button. The **Edit Rules** screen will be displayed. From here, the user can create the logic to archive cases based on a specific rule, i.e., archive case if the admission date is less than or equal to a specified date. See the Edit Rules (Rule Editor) segment in the Design section of the MedQuest Reference manual for a complete discussion of how to add/edit rules.

After completing the information for archiving cases, click on the <OK> button to begin archiving or the <CANCEL> button to cancel archiving. When all cases are archived, a popup message box will display the number of records added to the destination file, the number of duplicate records detected (duplicate



records are not added to the archive file), and the number of records deleted from the source file (which should be the same as the number of records added).

**DELETE CASE.** To delete a case, enter the case identification in the box provided or select the case from the cases list and then select the **Delete Case** function.

**NOTE:**

If the user enters a reabstraction case identifier, MedQuest will delete that reabstraction case.

**DELETE ALL CASES.** To delete all of the cases from the system, including the reabstraction cases, select the **Delete All Cases** function.

**NOTE:**

This function should only be used under system manager supervision.

**IQC FUNCTIONS.** Access the **MedQuest Quality** functions by clicking on the IQC menu item and selecting one of the following options: Single Compare and Load (Original/Reab); Load (Original/Reab); Single Compare (Original/Reab); and Select Option/Comparison Set. A valid case identifier and reabstraction ID must be selected/entered to access these functions.

Selecting the Single Compare and Load (Original/Reab) option will display the **Compare Cases (Single)** screen from the MedQuest Quality utility. From here, the user can compare the loaded cases or select other cases to compare. Clicking on the <OK> button will compare the selected cases. After the compare has been completed, click on the <CANCEL> button to load the MedQuest Quality Program.

Selecting the Load (Original/Reab) option will load the comparison into the MedQuest Quality utility.

Selecting the Single Compare (Original/Reab) option will display the **Compare Cases (Single)** screen from the MedQuest Quality utility. From here, the user can compare the loaded cases or select other cases to compare. Clicking on the <OK> button will compare the selected cases and then return the user to the MedQuest utility.

Selecting the Select Option/Comparison Set option will display the **Perform IQC Functions** screen. The user can select any of the options discussed above as well as the cases to compare.

**NOTE:**

The MedQuest Quality tool must be closed before the user can access the **IQC** functions from the MedQuest utility.

**NOTE:**

See the MedQuest Quality Reference Manual for a complete guide to the **IQC** functions.

**HELP.** Access the General Help by choosing this button. The General Help provides general MedQuest application help. Also available are General Help, Module Help, General Data Entry Guidelines, and About (MedQuest) from the HELP menu items. After compiling help from the dictionary, view that help file (XXXDICT.HLP where XXX is the module acronym) through Module help. The General Data Entry Guidelines is a place holder for the module developer to create this help file. About (MedQuest) identifies software and dictionary version information.

**EXIT MEDQUEST.** Exit MedQuest by selecting the <EXIT> button.

---

# MEDQUEST DATA ENTRY WINDOW

---

The **Data Entry** screen typically consists of the following components:

**TITLE BAR.** Displays the MedQuest function, **Data Entry**, and the current module.

**MENU BAR.** Displays the list of functions that can be performed on the **Data Entry** window. Each of these functions contains additional options that can be displayed using the mouse.

**DATA ENTRY BUTTON BAR.** Provides the different **Data Entry** functions. There are three functions available: **Enter Notes** (for a case), **Pause Data Entry** (stop data entry), and **Variable Assistant** (popup box displaying variables). Pausing a case will "stop the clock" that measures how long the user has been entering data. The variable assistant will also display any group of variables for viewing.

**DATA ENTRY TOP.** At the top of the **Data Entry** window, up to seven variables (and their values) can be displayed for quick reference. For example, the user may want to display the admission and discharge dates. The window top can be developed in the module design.

**TABS/SUBTABS.** Displays the names of the screens for the selected modules or the selected set of screens. Each module can have up to ten tabs. In addition, each tab can have up to five subtabs associated with it. When selecting the tab or subtab, the **Data Entry** window of the selected screen or subscreen will open in the space below it.

**DATA ENTRY WINDOW.** Contains variables that belong to the selected screen or subscreen. This is where data entry is performed.

---

## MEDQUEST CUSTOM DATA ENTRY WINDOW

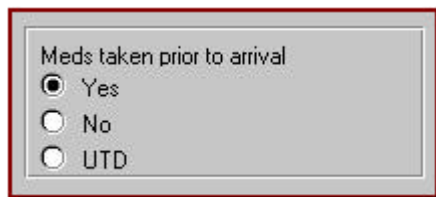
---

### Enter Data for a Variable

---

#### *Option (Pick One)*

---



The screenshot shows a small rectangular window with a light gray background. At the top, the text "Meds taken prior to arrival" is displayed. Below this text are three radio button options: "Yes" (which is selected, indicated by a filled circle), "No", and "UTD". The window has a thin black border and a slightly darker gray title bar area at the top.

The variable type **Option (Pick One)** includes the options from which only one item may be chosen. Choose an item by selecting the radio button or check box next to the item. If the variable contains more than a specified number of options (determined during module design), the options can be selected from a scrollable list box.

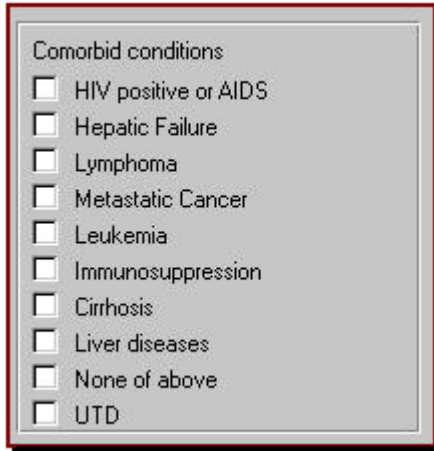
#### **NOTE:**

The system default is 15 options before the display changes to a list box. This value can be changed during design. The system has a maximum of 99 options before the variable is turned into a list box

by default. See the Preferences section for a complete description of the "Switch to a list box when number of items in Option (Pick One) or Option (Pick One or More) is greater than ?" option.

## Option (Pick One or More)

---



The screenshot shows a window titled 'Comorbid conditions'. Inside, there is a list of ten items, each preceded by an unchecked checkbox. The items are: HIV positive or AIDS, Hepatic Failure, Lymphoma, Metastatic Cancer, Leukemia, Immunosuppression, Cirrhosis, Liver diseases, None of above, and UTD. The window has a standard Mac OS-style title bar and a scroll bar on the right side.

The variable type **Option (Pick One or More)** includes a list of options from which one or more items can be chosen. Choose an item(s) by marking the check box(es) next to the desired item. If the variable contains more than a specified number of options (determined during module design), the options can be selected from a scrollable list box.

### NOTE:

The system default is 15 options before the display changes to a list box. This value can be changed during design. The system has a maximum of 99 options before the variable is turned into a list box by default. See the Preferences section for a complete description of the "Switch to a list box when number of items in Option (Pick One) or Option (Pick One or More) is greater than ?" option.

### NOTE:

Note that the last item on the list that clears all the other items is optional. See the Design section for a complete description of the "Clear Others If True" option.

### NOTE:

Note that each item on the list can be marked as "UTD." Clicking on the option with the right mouse button will display the Mark as UTD option. Selecting this option will show a "low highlight check" in the option box. See the Preferences section for a complete description of this feature.

## Option (External List)

---

A variable defined as variable type **Option (External List)** includes a list of options from which one or more items can be selected. The options are derived from an external database provided by the data entry system designer.

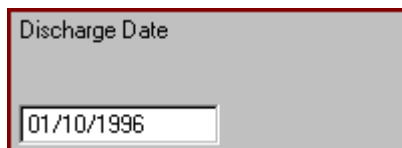
If during the design process an **Option (External List)** variable has been linked to another variable, the **Option (External List)** variable will remain blank (except for UTD if that option has been included) until the linked variable has been entered. For instance, to link a **State** variable with a **County** variable, the **State** variable would be entered first so that the **County** variable would filter (display) only those counties associated with the selected state.

**NOTE:**

If "UTD" is included as an option, it will appear as the first item on the list and selecting "UTD" will clear all the other selected items on the list.

## *Date*

---

A screenshot of a software interface showing a label 'Discharge Date' above a text input field. The input field contains the date '01/10/1996' in a standard MM/DD/YYYY format.

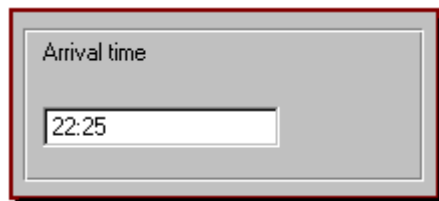
The variable type **Date** accepts data entered in the date format that has been specified in the design (the default format for date is MM/DD/YYYY). An additional option available for the Date and Date/Time variables is the capability to save a date using only the year, which is a Year Acceptable option, or the month and the year, which is a Year/Month Acceptable option. These options are chosen during the design process. This allows the user to save a partial date instead of marking the date as "UTD."

Use the space bar to enter blank spaces for the month and day or just the day, depending on the option(s) chosen during design.

If the date cannot be determined, press the hot key combination **<SHIFT>** and **<?>** (**<SHIFT>** and question mark keys).

## *Time*

---

A screenshot of a software interface showing a label 'Arrival time' above a text input field. The input field contains the time '22:25' in a HH:MM format.

The variable type **Time** accepts data entered in the time format specified in the design (default is the military time format beginning with 00:00 (midnight) and ending with 23:59 (11:59pm)).

If the required value for a time entry field cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## *Date/Time*

---

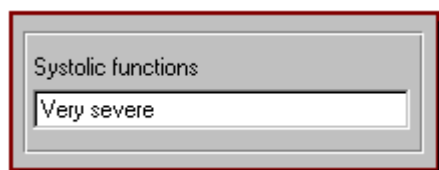
The variable type **Date/Time** requires the data abstractor to enter the date/time format that has been specified during the design. An additional option available for the Date and Date/Time variables is the capability to save a date using only the year, which is a Year Acceptable option, or the month and the year, which is a Year/Month Acceptable option. These options are chosen during the design process. This would allow the user to save a partial date instead of marking the date as "UTD."

Use the space bar to enter blank spaces for the month and day or just the day, depending on the option(s) chosen during design.

If the required value for a date/time entry field cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## String

---

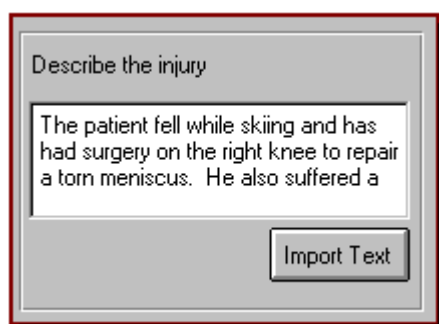
A screenshot of a data entry window for a 'String' variable. The window has a title bar and a label 'Systolic functions' above a single-line text box. The text box contains the text 'Very severe'.

The variable type **String** accepts up to 99 characters in any alphanumeric combination. If a format is specified in the design (i.e., only dashes and parentheses allowed in certain positions in the string), the data abstractor will be required to enter the data in that format.

If the value cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## Memo

---

A screenshot of a data entry window for a 'Memo' variable. The window has a title bar and a label 'Describe the injury' above a multi-line text box. The text box contains the text 'The patient fell while skiing and has had surgery on the right knee to repair a torn meniscus. He also suffered a'. Below the text box is a button labeled 'Import Text'.

The variable type **Memo** accepts up to 32,000 characters in any alphanumeric combination. The data can be entered directly into the text box or imported from a file by using the **<IMPORT TEXT>** button.

If the value cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## Memo (ICD9)

---

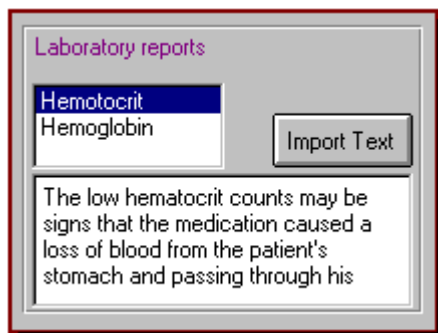
The variable type **Memo (ICD9)** accepts up to 32,000 characters in any alphanumeric combination. In addition to text entry, this variable type provides two applicable assistive data entry buttons (i.e., **<ICD9 DX>** and **<ICD9 PX>**) for retrieving ICD-9-CM Diagnosis Codes and ICD-9-CM Procedure Codes. The buttons provide access to the **Select Codes** window from which the data abstractor can view a list of codes belonging to the group selected, search for a code by either the code or the code description, select a desired code(s), and add the code(s) to the memo field.

If the option to include a decimal point associated with an ICD9 code is selected during the design process, it will be displayed on the **Data Entry** screen after a selection has been made. Also, a code originally saved as 30089 in the datastore database will now be saved as 300.89.

If the value cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## Memo (List of)

---



Laboratory reports

Hematocrit  
Hemoglobin

Import Text

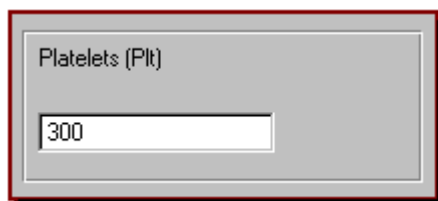
The low hematocrit counts may be signs that the medication caused a loss of blood from the patient's stomach and passing through his

The variable type **Memo (List of)** displays a text box into which the data abstractor can enter data pertaining to the selected item on the list above it. The data can be entered directly into the text box or imported from an ASCII file by using the <IMPORT TEXT> button.

If the value cannot be determined, press the hot key combination <SHIFT> and <?>.

## Number (No Units)

---



Platelets (Plt)

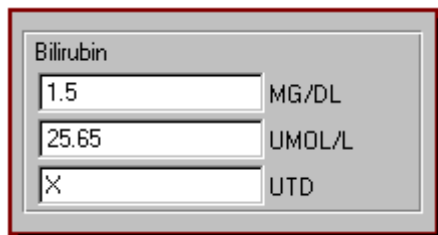
300

The variable type **Number (No Units)** accepts data entered in numeric format, including one decimal point, if applicable. If a format is specified in the design, the data abstractor will be required to enter the data in that format.

If the value cannot be determined, press the hot key combination <SHIFT> and <?>.

## Number (Units)

---



Bilirubin

1.5 MG/DL

25.65 UMOL/L

X UTD

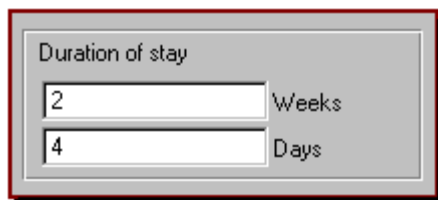
The variable type **Number (Units)** accepts a numeric entry including decimal points, if applicable, in one of the measurement units defined on the screen. Regardless of the unit entered, the application converts the value entered to the base unit which is the first item on the list.

If a UTD unit box is present and the unit cannot be determined, enter the value in the UTD box. When the value is saved, a "U" is appended to the end of the string.

If the value cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## Number Set

---



The variable type **Number Set** accepts data entered in numeric format in one or all of the measurement units defined on the screen.

If the value cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## ICD9 Code

---

The variable type **ICD9 Code** allows the data abstractor to select either an ICD-9-CM Diagnosis Code or an ICD-9-CM Procedure Code from the search window accessible through the assistive buttons (i.e., **<ICD9 DX>** and **<ICD9 PX>**). The buttons provide access to the **Select Codes** window from which the data abstractor can view a list of codes belonging to the group selected, search for a code by either the code or the code description, select a desired code(s), and add the selected code to the variable. Only one code is saved.

If the option to include a decimal point associated with an ICD9 code is selected during the design process, it will be displayed on the **Data Entry** screen after a selection has been made. Also, a code originally saved as 30089 in the datastore database will now be saved as 300.89.

If the value cannot be determined, press the hot key combination **<SHIFT>** and **<?>**.

## Diagram

---

The variable type **Diagram** lists diagrams that the data abstractor can select, mark, and write comments about. On the **Mark Diagram/Comment** screen selected, the data abstractor can use assistive devices located at the top left corner of the screen to mark and write descriptions about the selected images. The assistive devices include:

- **<LINE> Button.** Click on this button to create a straight line.
- **<SQUARES> Button.** Click on this button to create a square.
- **<CIRCLE> Button.** Click on this button to create a circle.
- **<ARROW> Button.** Click on this button to create a line with an arrow at the end.
- **<A> Button.** Click on this button to place text on the image. After entering the text, click on the **<OK>** button. Place the cursor where the text is to be dropped and press the left mouse button.
- **<ERASER> Button.** Select the object to delete from the *Select Objects to Delete* list box and click on the **<ERASER>** button.
- **<COLORS> Button.** Click on this button to select the color of the object(s) being created.

## Label

---

This variable type does not require data entry.

## No UTD Option

---

When this feature is selected during the design process, the data abstractor will not be able to enter the "UTD" option (<SHIFT> + ?) for a variable. This feature applies to variable types: Option (External List), Date, Time, Date/Time, String, Memo, Memo (ICD9), Memo (list of), Number (no units), Number (units), ICD9 Code, and Diagram.

## Data Entry Assistive Aids

---

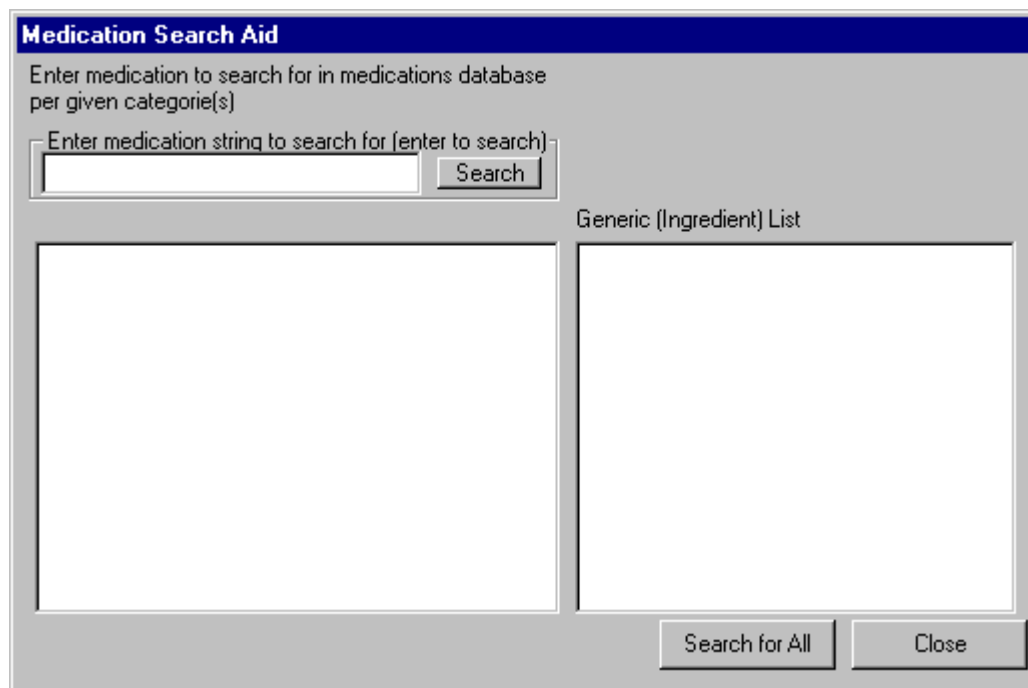
There are two assistive aids available during data entry: a medication search aid and a feet/inches calculator.

### Medication Search Aid

---

During design, the developer may have questions about various medications that only require a yes/no type of response. Since medications are only displayed in a grid variable, the abstractor would not have a list from which to view. Therefore, a search aid for medications has been added.

For example, a question asking if Warfarin was given (yes/no) is included as a variable on the **Medications** screen. The data abstractor can right mouse click on that variable and select the Medication Search Aid option. A popup search aid is displayed, as shown below.



The image shows a "Medication Search Aid" dialog box. It has a title bar with the text "Medication Search Aid". Inside the dialog, there is a text area with the instruction "Enter medication to search for in medications database per given categorie[s]". Below this, there is a text input field with the placeholder text "Enter medication string to search for (enter to search)" and a "Search" button to its right. To the right of the input field, there is a label "Generic (Ingredient) List" above a large empty rectangular box. At the bottom of the dialog, there are two buttons: "Search for All" and "Close".

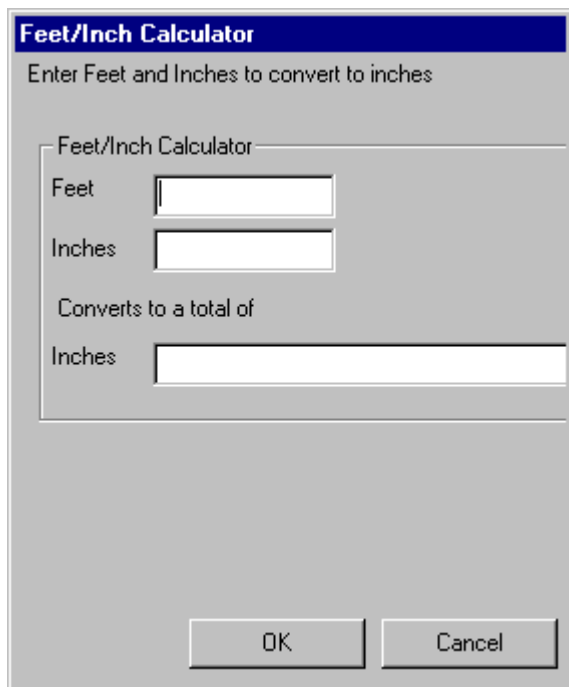


The abstractor can type in a letter or string of letters and click on the <SEARCH> button to display a list of medications or click on the <SEARCH FOR ALL> button to display a list of medications (specified during design for Warfarin).

## ***Feet-Inches Calculator Aid***

---

During design, the developer may wish to create a variable for height in inches and feet/inches. By adding the popup search aid for calculating feet/inches during design, the data abstractor can right mouse click on that variable(s) and select the Feet-Inches Calculator option. A popup search aid is displayed, as shown below.

A screenshot of a Windows-style dialog box titled "Feet/Inch Calculator". The title bar is dark blue with white text. Below the title bar, the text "Enter Feet and Inches to convert to inches" is displayed. The dialog box has a light gray background. Inside, there is a section titled "Feet/Inch Calculator" with a thin border. This section contains three input fields: "Feet" with a small vertical cursor, "Inches", and "Converts to a total of Inches". At the bottom of the dialog box, there are two buttons: "OK" and "Cancel".

The search aid enables the abstractor to enter the feet/inches and calculate the total inches, or enter the inches to calculate the total feet/inches.

## **Enter Data in a Grid Variable**

---

There are two types of grids: a medications grid and an incident grid. A data abstractor can enter more than one row of data for each variable added to the grid, as described below.

### ***Enter Data in a Medication Grid***

---

The medication grid collects a drug name in the first column of the grid. If there are any additional variables (e.g., the start date of the medication) that have been designed in the grid, those variables will appear in the columns to the right of the first column.



### Advanced User: Changes in the Medications Database Structures

Beginning with the release of MedQuest version 4.10, designers of new modules will have the flexibility of selecting the old MedQuest Medications database or the Multum Medisource™ Lexicon Medications database for use during data entry. Any modules designed with the MedQuest software versions prior to 4.10 will be required to use the old MedQuest Medications database if they include medication grid variables. However, this database will no longer be formally maintained. The major difference between the two databases is how the tables are linked.



### Advanced User: Data Entry Using a Custom Medications List

The text boxes used to display the medication selected from the pick list will both display the same name. Because the allowable medications were preselected during the design process, both brand names and generic names can be included in the custom list. Since these are stored in separate tables in the data dictionary, this bypasses the need to perform a lookup in the medications database during data entry.

## Add a Medication Using the Old MedQuest Medications Database

---

To add a medication, type the medication name into the box beneath the grid. Once enough characters have been entered, press the <ENTER> key or use the **Search** function to locate the medications that match the string entered. The **Search** function is not necessary if the entire name was typed. If a medication in the search list is the one to save, select that medication and then the **Add** function to add that medication to the grid. There is no limit to the number of medications that can be added to the grid.

Note that for collecting only medications of a certain group (e.g., all antibiotics) or a custom meds list (designated during design), a list of the medications associated with that group or specified list can be built by using the **List Applicable Medications** function during data entry. This will take some time because the request has to be processed against all of the records in the Medications database. A medication from this list can be selected to save as well. Selecting a medication from either search list will also display the generic name of that medication.

When selecting the **Add** function, there are a number of rules that MedQuest will check:

**MEDICATION NOT IN DATABASE.** This warning will be displayed if the medication or ingredient is not in the database. The data abstractor can accept the medication anyway.

**MEDICATION NOT OF TYPE EXPECTED.** If the data abstractor is required to enter a medication of a certain group (e.g., an antibiotic), he/she will be warned if the medication entered is not of that type. The data abstractor can bypass the warning.

**DUPLICATE GENERIC.** If the data abstractor tries to enter a duplicate generic, he/she will be prevented from doing so, if this optional rule type is selected in the design.

**DUPLICATE GENERIC, ACCEPT VALUE.** If the data abstractor tries to enter a duplicate generic, he/she will be warned but will be able to bypass the warning and accept the value, if this optional rule type is selected in the design.

**NO GENERIC WARNING.** If this optional rule type is selected in the design, the data abstractor will not receive any warning for a duplicate generic.

## Add a Medication Using the Multum Medisource™ Lexicon Medications Database

---

To add a medication, type the medication name into the box beneath the grid. Once enough characters have been entered, press the <ENTER> key or use the **Search** function to locate the medications that match the string entered. The **Search** function is not necessary if the entire name was typed. If a medication in the search list is the one to save, select that medication and then the **Add** function to add that medication to the grid. There is no limit to the number of medications that can be added to the grid.

A search list, an ingredients list, and the old MedQuest Medications Database list (if this option was selected during design) are displayed on the screen as well.

Note that to collect only medications of a certain group (e.g., all antibiotics) or a custom meds list (designated during design), a list of the medications associated with that group or specified list can be built by using the **List Applicable Medications** function during data entry. This will take some time because the request has to be processed against all of the records in the Medications database. A medication from this list can be selected to save as well.

**MEDICATION NOT IN DATABASE.** This warning will be displayed if the medication or ingredient is not in the database. The data abstractor can accept the medication anyway.

**MEDICATION NOT OF TYPE EXPECTED.** If the data abstractor is required to enter a medication of a certain group (e.g., an antibiotic), he/she will be warned if the medication entered is not of that type. The data abstractor can bypass the warning.

**DUPLICATE GENERIC NAME.** If the data abstractor tries to enter a duplicate generic name, he/she will be prevented from doing so, if this optional rule type is selected in the design.

**DUPLICATE GENERIC NAME, ACCEPT VALUE.** If the data abstractor tries to enter a duplicate generic name, he/she will be warned but will be able to bypass the warning and accept the value, if this optional rule type is selected in the design.

**NO GENERIC NAME WARNING.** If this optional rule type is selected in the design, the data abstractor will not receive any warning for a duplicate generic name.



### Advanced User: The Multum Medisource™ Lexicon Medications Database

The Multum Medisource™ Lexicon Medications database has been modified from its original form for use with MedQuest. When downloaded from the Multum homepage, it is about 35 MB in size. Tables that are not useful for data entry have been dropped, and a table was created linking brand names with the Multum Medisource™ Lexicon drug categories.

## Delete a Medication

---

To delete a medication and related data in that row, first select the row to delete (choose the first column of that row) and then select the **Delete** function. MedQuest will ask for verification of the deletion before performing this function. The data cannot be recovered after the row has been deleted.

## Add Data to Related Variables

---

If the grid contains related variables that must be answered (e.g., the start date of the medication), enter the value for each variable by selecting that cell in the grid. The cells where an answer is expected are

the cells that have a white background. The cells that require no data entry, based on the rules of entry established in the design, are gray.

When entering data into a cell, a popup box for the appropriate variable type will appear. Enter the data into that popup box. For example, for the start date of the medication, enter a date into the popup box. The popup box displays the identical data entry format as displayed in the normal **Data Entry** window. If necessary, resize the popup box by using the mouse to drag the edges of the box. Refer to the discussion of how data are entered for the different variable types.

**LIMIT THE NUMBER OF ROWS.** If the module has been designed with a limit on the number of rows that can be added to a medication grid, the system will issue a warning that the maximum number of rows has been reached. No additional rows can be added.

**NO ADD BUTTON.** If there is no <ADD> button, the data abstractor will not be able to add any rows to the grid.

**NO DELETE BUTTON.** If there is no <DELETE> button, the data abstractor will be unable to delete any rows from the grid.

## ***Enter Data in an Incident Grid***

---

The first column of an incident grid is a read-only counter created when a row is added to the grid. There should always be additional variables in the grid that are associated with this counter (e.g., for a procedure, the start date of the procedure and the procedure type). These variables (columns) appear to the right of the first column in the grid.

### **Add an Incident**

---

Up to 9999 rows of data (i.e., 9999 incidents) can be added into a grid. To add an incident, select the **Add** function. This increments the grid counter by 1. If the designer has limited the number of incidents that are allowed, only the designated amount can be added.

### **Delete an Incident**

---

To delete a row of data, first select the row to be deleted (highlight the first column of that row), then select the **Delete** function. MedQuest will ask for verification of the deletion before performing this function. The data cannot be recovered after the row has been deleted.

### **Add Data to Related Variables**

---

If an incident contains related variables that must be answered (e.g., the start date of a surgical procedure), enter the value for that variable for each incident by selecting that cell in the grid.

When selecting a cell, a popup box for the appropriate variable type will appear. Enter the data into that popup box. For example, for the start date of a surgical procedure, enter a date into the popup box. The popup box displays the identical data entry format as displayed in the normal **Data Entry** window. If necessary, resize the popup box by using the mouse. Refer to the discussion of how data are entered for the various variable types.

**PREVENT ENTRY OF DUPLICATES.** The system provides an option to prevent entry of duplicate records having the same key-link fields. Key-link fields are primary fields used to match incident records in the MedQuest Quality (IQC) program. The system issues a warning if an incident has been entered

that has a duplicate key-link field. However, this warning can be overridden and the system will accept the duplicate entry.

**LIMIT THE NUMBER OF ROWS.** If the module has been designed with a limit on the number of rows that can be added to the incident grid, the system will issue a warning that the maximum number of rows has been reached. The system will not allow any additional rows to be added.

**PRELOAD THE NUMBER OF ROWS.** If the module has been designed with the **Preload** feature, when the grid variable is opened a predefined number of rows will be added.

**NO ADD BUTTON.** If there is no <ADD> button, the data abstractor will not be able to add any rows to the grid. This might be used when the grid has been preloaded with a specified number of data rows.

**NO DELETE BUTTON.** If there is no <DELETE> button, the data abstractor will not be able to delete any rows from the grid. This might be used when the grid has been preloaded with a specified number of data rows.

## Special Features

---

### *Note Box*

---

To enter a general note about a variable during data entry, select the <NOTES> button. This feature allows the user to enter a note containing up to 32,000 alphanumeric characters. To insert the screen title of the current variable at the bottom of the note, select the <COPY> button. Since all the notes for a given abstraction case are stored in one file, copying the variable screen title before writing a note provides a good reference point to the variable.

### *Pause Button*

---

Selecting the <PAUSE> button will stop the internal MedQuest clock that keeps track of the amount of time a case has been abstracted.

### *Variable Assistant Popup*

---

When selecting the variable assistant popup from the top of the **Data Entry** window, a list of variables that have been specified in the design will appear with their associated values. This lets the user quickly view the values of certain variables, e.g., the user may want to display a set of key dates/times together in the assistant popup.

### *Clear Value on a Variable Radio Button*

---

Clicking on the <RIGHT MOUSE> button and selecting the Clear Variable option can clear variables with radio buttons.

### *Short Cut Data Entry*

---

This feature is available by selecting the Enter Data On Another Screen option from the OPTIONS menu during data entry. This allows the abstractor to enter data for a specified variable on another screen while remaining on the selected screen. This option is not available for **Sequential Data Entry**.

## ***Printer Setup***

---

This option lets the user setup his/her printer options. Select a printer and define the document properties (e.g., paper size, etc.).

## ***Print Case Summary Report***

---

This option gives the user the capability of printing the **Case Summary** report for the current case. From the FILE menu item, select the Print Case Summary Report option. The **View/Print Reports** screen will be displayed. The **Case Summary** report is the only report available and will display, save, or print the currently selected case. See the Reports section for a complete discussion of the **Report** function.

---

## **CLINICAL HELP (DATA ENTRY)**

---

### **View Clinical Help**

---

View the clinical help for a variable in the main **Data Entry** screen by clicking the right mouse button anywhere in the selected variable's frame. To view the clinical help during data entry in a grid, double click the left mouse button on the column title.

When viewing the clinical help that has not been compiled, the help will appear in the same format that it was entered during the design. If the help has been compiled, then it will appear as a Microsoft help file. In this format, the help appears as a separate help window. With a Microsoft help file, the Microsoft help view program supports a number of functions. See Microsoft Windows help (WINHELP.HLP) for more details.

---

## **DATA ENTRY RULES (DATA ENTRY)**

---

### **Execute Data Entry Rules**

---

Any data entry rules that have been built into the dictionary will be executed by specific events that occur during data entry. The events that can invoke rules include:

- On Loading a Case
- On Loading a Screen
- On Entry of a Variable
- After Exiting a Case
- After Leaving a Screen
- After Entry of a Variable
- After Entry of a Grid

**NOTE:**

Rules to be executed after exiting a case will only execute if the post processing after exiting a case is enabled. Rules to be executed after leaving a screen will only execute if the post processing after a screen is enabled.

## Enable Variable Rule

---

This rule is executed on entry of a variable, on loading a screen, and on exiting a case. When the result of the rule logic is true, the variable on the screen is enabled; otherwise it is disabled. A rule can also be created that enables a variable on a screen other than the one being entered.

The **Enable Rule** is used for two purposes. First, while performing data entry, this rule type enables (if true) or disables (if false) the variable for entry. Second, after finishing entry on a screen or a grid, or attempting to exit a case, a warning will appear (if the post processing is turned on) if there is a mandatory variable that is enabled (a chance to enter data) and it does not have a value entered.

**NOTE:**

An indented variable that has no rule associated with it will always be disabled.

## Disable Variable Rule

---

This rule is executed on entry of a variable, on loading a screen, and on exiting a case. When the result of the rule logic is true, the variable on the screen is disabled; otherwise it is enabled. A rule can be created that disables a variable on a screen other than the one being entered.

The **Disable Rule** is used for two purposes. First, while performing data entry, this rule type will disable (if true) or enable (if false) the variable for entry. Second, after finishing data entry on a screen or a grid, or attempting to exit a case, a warning will appear (if post processing is turned on) if there is a mandatory variable that is not disabled (a chance to enter data) and it does not have a value entered.

**NOTE:**

An indented variable that has no rule associated with it will always be disabled.

## Skip to Rule

---

This rule is executed after entry of a variable. The rule will execute when the "action" variable's value is changed. If the rule is true, then the variables between the "action" variable and the "skip to" variable will be disabled.

**NOTE:**

The user cannot skip to a variable that occurs prior to the variable being skipped.

## Hot Key Rule

---

This rule is executed after entry of a variable. It is used to save a value from another variable to the one being entered when a hot key is pressed. For example, use this rule type to copy the admission date from a variable in the header that was downloaded, perhaps to assist with verifying the correct entry of that variable.

**NOTE:**

Choose the designated "Hot Key" to invoke the save action.

## After Entry Warning Rule

---

This rule is executed after entry of a variable. If the rule is true, a warning message will be presented (an English translation of the rule logic). For example, a common warning message is that the value of the variable is not within the specified range (e.g., temperature is > 120 degrees). It is possible to bypass all warnings by selecting the <OK> button; or, by selecting the <RETRY> button, to change the variable.

**NOTE:**

If the Retry Only option was chosen during the design, the data abstractor will not be able to bypass the warning.

## Save Value to Variable Rule

---

This rule is executed after entry of a variable. In some circumstances, it is necessary for the value of one variable to cause a selected value to be saved to another variable, e.g., to save "0" to lowest blood pressure if the patient expires. These rules will be evident only during data entry by performing data entry on the variable that has been changed by the rule. If the rule causes the value to be over written, the data entered will be saved in the database unless the rule is executed again.

## Derive If Rule

---

When entering a value that is included in a **Derive If** rule, the derived variable will be calculated when the focus is lost for the variable being entered (i.e., either when another variable or screen is chosen, or the case is exited). If a null value is included, the derived variable will be null.

## Exit Case Rule

---

This rule is executed on loading a screen and after entry of a variable. There are certain values or combinations of values that can require exiting data entry mode and marking the case as complete. These rules are constantly checked to ensure that as soon as there is an exit condition the data abstractor is warned to exit the case.

**NOTE:**

The **Exit Case** rule does not prevent changing values, it merely provides a warning that the user should exit the case. This is so he/she can correct a value that might have been entered incorrectly forcing the case to be exited.

## Post Processor Warning Rule

---

This rule is executed (if post processing is enabled) after leaving a screen or a grid, and upon exiting a case. The **Post Processor Rules** provide a warning (English language translation of the rule logic) and a message suggesting the action to be taken. The user can ignore these warnings without having any impact on the data entry or the ability to mark a case as complete. These warnings appear in the **Post Processor Warnings** window.



## Post Processor Mandatory Override

---

This rule type is executed (if post processing is enabled) to override a mandatory warning in post processing for a given variable (after entry of a screen or after entry of a case).

### EXAMPLE:

The rule indicates that if the **Discharge Disposition** variable is answered "Patient expired," do not answer the mandatory **Discharge BP** variable. By assigning the Post Processor Mandatory Override Rule to the mandatory **Discharge BP** variable, when the answer to **Discharge Disposition** is "Patient expired," the **Discharge BP** does not require data entry.

## View Post Processor Warnings

---

There are several warnings that occur in the **View Post Processor Warnings** window. This window appears (if the Post Processor Warnings option was turned on during the design) when completing entry of a screen and/or exiting a case.

All the warnings appear in a list. The details of a given warning can be viewed by clicking on it. Continue either by pressing the <OK> button to complete the screen and then exit the case, or by pressing the <CANCEL> button to return to data entry.

There are two primary types of warnings. The first, and most important, is the display of those variables that are marked as mandatory. These variables require a value. Return to the **Data Entry** screen to complete entry of these variables before marking a case as complete.

The second type of warning is a message which describes a "hint" or "tip" for data entry. For example, it might be useful to know that the diagnosis code suggests ischemia yet the answer for a question on "Ischemia Present" is "No." These types of warnings can be ignored during data entry.

## Disable Screen Rule

---

The **Disable Screen** rule is used to prevent a data abstractor from performing data entry on a given screen until certain conditions are met. The variables on that screen are disabled only if the rule executes as true. (By default, the variables on a screen are enabled when that screen is loaded.) Note, this rule type is not available for **Sequential Data Entry**.

### NOTE:

When the **Disable Screen** rule is true, all values entered will be cleared. Therefore, if an abstractor enters data and then changes an answer that causes the disable rule to execute to true, all data affected by this rule will be deleted.

## Do Not Load Screen Rule

---

The **Do Not Load Screen** rule is used to prevent a data abstractor from performing data entry on a given screen until certain conditions are met.

Typically, this rule is used to force the abstractor to enter data on screens in a certain order or to insure that the abstractor will not enter data on other screens until certain key screens are completed first. Note, this rule type is not available for **Sequential Data Entry**.



### Advanced User: Do Not Load Screen Rules Versus Disable Screen Rules

The **Do Not Load Screen** rule does not disable that screen. It simply prevents a given screen from being loaded under certain conditions. If there are mandatory variables on a screen which the user cannot load, he/she will not be able to complete the case in data entry (post processor errors will exist). If the variables on a screen should be disabled for a given set of logic, then use the **Disable Screen** rule.

## On Entering Case Load Screen Set ? If.. Rule

---

The **On Entering Case Load Screen Set ? IF..** rule is executed when a case is loaded for data entry. If any of the rules are true, then the screen set for the first rule that executes as true is used for data entry. If none of the rules are true, then the first screen set in the list is used by default.

---

## EXIT DATA ENTRY

---

### Post Processing After Exit

---

If the option to run the Post Processor on exiting a case is selected in the design, MedQuest will check all of the post processor rules when attempting to exit a case. Mandatory rules will always be checked. If a variable is marked as mandatory, but has not been entered, and is not disabled, a warning will appear that this variable is mandatory. If any mandatory warnings exist, the case cannot be marked as complete.

In addition, any post processor warnings will be displayed if the logic evaluates to true. These warnings are informational only and do not have to be resolved before a case can be marked as complete.

Finally, when exiting a case, all of the derived variable rules will execute. This ensures that the derived variables will all have a recalculation done consistently.

If the post processing is not turned on, no warnings will appear.

## Case Status Window

---

If the design option to display the **Case Status** screen has been selected, the status of the case will need to be marked upon exiting the case. At a minimum, the abstractor will be required to mark the case as still in use or as completed. These are the default options on the **Case Status** screen.

In addition, if during the design other options or variables have been added to the **Case Status** screen, these other options or variables can be marked depending on the enable/disable rules that have been put into place.

To mark the status, select the best choice (and answer any additional questions that appear) and press the <OK> button. This will complete the data entry on the case.

---

## MEDQUEST DATA ENTRY ENGINE

---

The MedQuest Data Entry Engine is a separate program from the MedQuest program. It contains only the **Data Entry** functions associated with MedQuest and does not include any of the **Design** functions. Since the MedQuest Data Entry Engine program contains only the code necessary to run data entry, its performance speed is increased, which is important for data abstraction.

**NOTE:**

The only report available in the MedQuest Data Entry Engine is the **Case Summary Report**.

**NOTE:**

See the Select Project Options segment in the Module Management section for additional features.

---

# REPORTS

---

---

## VIEW/PRINT REPORTS

---

MedQuest provides the user with a number of reports that can be used to facilitate the collaborative data entry system design process and the quality review of the abstracted data.

The clinical data dictionary contains all of the information necessary to use the data entry system that has been designed. At any point during the design process, the variables may be documented to provide a record of the data elements and their attributes. The document generated from the data dictionary during development helps facilitate communication among team members and tracks changes at different points in the process. Viewing reports of the individual variables or comparing one version of the data dictionary to another version helps identify the differences. Standard MedQuest reports can be an asset in facilitating the collaborative process.

The data dictionary reports present the key elements of the data dictionary without requiring that the development team members have the knowledge of the technical structure of the data dictionary. The reports can facilitate team communication by displaying the data elements along with their attributes in a concise and readable format. It is not necessary for a team member to understand all of the relationships among all variables since a report can display the essential elements of the data dictionary in a format that can be shared among various individuals. This enables all team members to collaborate using a single, standardized report or set of reports that contain certain elements of each variable.

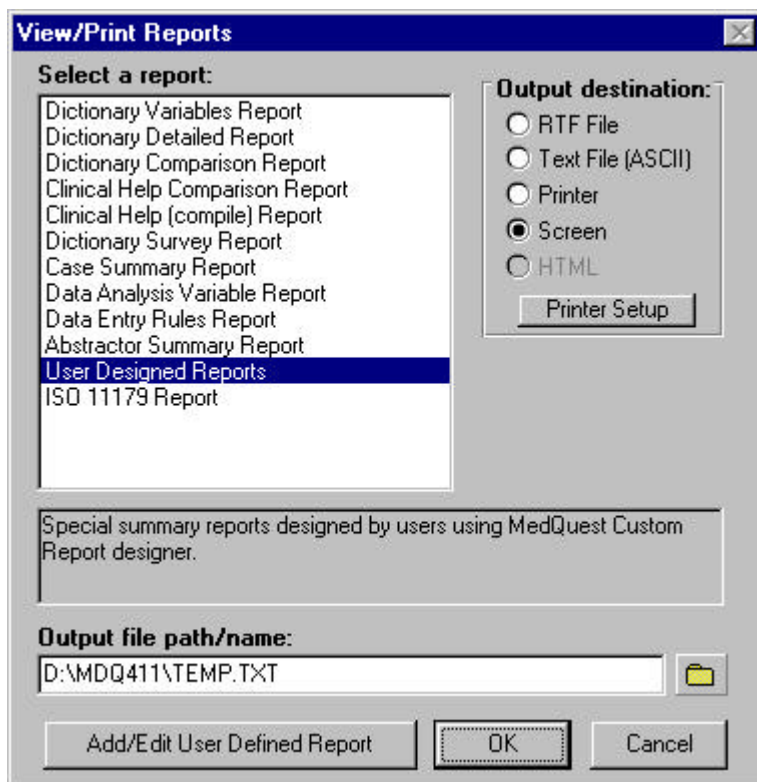
Another aspect of system development is to validate whether the question, as defined, provides the type of clinical data that is required for analysis. This is often accomplished by reviewing medical records in the field. At this stage, it may be desirable for the analyst to conduct the review using a paper instrument rather than a computer. The data can be entered into the data entry system at a later time, if necessary.

---

## SELECTING A REPORT

---

Select the **Reports** function by clicking on the <VIEW/PRINT REPORTS> button or by selecting the View/Print Reports option from the OPTIONS menu item on the **Cases** screen. The **View/Print Reports** window will display the various types of reports that can be produced.



Reports can be produced in five formats:

- Rich Text Format (RTF) File, for importing into a word processor;
- Text File;
- Report to the Printer;
- Report to the Screen; and
- HTML (currently not available).

For each of these modes, a temporary file is always produced. Therefore, it is necessary to specify the location and the name of the file to change the default file location and the file name provided by MedQuest. Otherwise, the files will overwrite each other.

If the user sends a report to the printer, the option to setup the printer using the function provided is available.

After selecting the report, select the <OK> button to begin generating the report or, if applicable, go to the report specific selection criteria associated with the report selected.

## Dictionary Variables Report

This report lists all the selected module variables sorted by the **Short title**. The report also includes the 8-character field name (**Fieldname**), the module acronym, and the type of variable. All mandatory variables will be denoted by an asterisk (\*).

When this report is selected, the option is to sort the variables by natural order by screen or alphabetically by short title.

## Dictionary Detailed Report

---

This report displays the detailed information that was entered into the dictionary database during the design process. When this report type is selected, a set of options is available from which to select:

**ALL SCREENS.** This option reports all of the variables on all of the screens.

**SELECT SCREEN SETS/SCREENS.** This option is used to select specific screens on which to report by selecting the screen set and screens that are in that set. Note, at least one screen must be selected if this option is chosen.

**INCLUDE CLINICAL HELP.** This option reports all of the variables on the selected screens and includes their clinical help.

## Dictionary Comparison Report

---

This report provides a comparison between any two dictionaries of the same module. Often in the design process, it will be necessary to compare a previous version of the dictionary with the current version. To obtain a meaningful result from the comparison, it is recommended that a new directory be created to store the point-in-time dictionary. To perform the comparison, indicate the locations and the file names of the current module and the previous module.

When this report is produced, select any or all of the tables associated with the dictionary for comparison. By using the option of selecting specific tables, the changes to a particular area in the dictionary can be reviewed. For instance, if the Equations table is selected, any differences in the data entry rules will be listed in the comparison report.

The data dictionary tables include:

- Variables in data entry system (MASTER)
- Screens in data entry system (SCREENS)
- Sets of screens in data entry system (SCREENSETS)
- Variables on screen(s) (VIEWS)
- Variables on grids (MULTVIEWS)
- System data (SYSTEM)
- Data entry rules (EQUATIONS)
- Classification types (CLASSES)
- Variable classifications (FIELDCLASSES)
- Numeric conversion groups (tblConvertGroups)
- Numeric conversion units (tblConvertUnits)

- Numeric conversion factors (tblConvertFactors)

## Clinical Help Comparison Report

---

This report compares the clinical help for a previous version of the dictionary with the current version. To obtain a meaningful result from the comparison, it is recommended that a new directory be created to store the point-in-time dictionary. To perform the comparison, indicate the locations and the file names of the current module and the previous module.

When this report is produced, the variables whose help is different will be displayed.

## Clinical Help (Compile) Report

---

The clinical help entered into MedQuest for the screens and variables can be formatted into a word processing compatible file and compiled into a Microsoft Windows help file (HLP file). The latter type of file can be produced only by using the Microsoft Help compiler (not provided with the MedQuest software) version 3.10.0 or later. This compiler (HCP.EXE) is commonly delivered with tools that allow development of a Microsoft help file or can be obtained from Microsoft or other Internet sites.

When selecting the option to compile help, the location of the compiler (HCP.EXE) program must be specified. Enter the path to this program to compile help.

While entering clinical help, comments that are associated with the help can also be entered. These comments can be included in this report (and the compiled help) by selecting the option provided.



### Advanced User: Using Help Report

The most common use of the **Clinical Help** report is to produce a word processing document that contains all of the help being built for a system. This help is often the most complex part of a data entry system and, depending on its size, may take the most time to enter.

After producing this report (RTF report), the report can then be opened in a word processor (e.g., WordPerfect or Word) and additional features such as a cover page can be added. Since the report utilizes standard WordPerfect/Word styles to format the document, a table of contents can be generated. This document is very useful for communicating with a clinical person about the contents of the dictionary.

The compiled help file is a hypertext file that can be viewed by any user with Windows. This is very advantageous for sending (via the Internet or other ways) the dictionary description to others.

## Dictionary Survey Report

---

This report produces a data entry instrument that can be used to collect clinical data on paper. The report is a replication of the data entry system. Copies of this blank report may be sent to users who do not have access to or are not trained in using the MedQuest data entry system. When selecting this report type, a set of options is available from which to choose:

**ALL SCREENS.** This report lists all of the variables on all of the screens.

**SELECT SCREEN SETS/SCREENS.** This report lists variables on the selected screen sets and screens. At least one screen must be selected if this option is chosen.

**NOTE:**

If the Number Variables option in the PREFERENCES menu has been selected during design, the variables in the **Dictionary Survey** report will be numbered sequentially.

## Case Summary Report

---

This report lists the data that have been entered for a case(s) by a data abstractor. The report for a case can be produced by typing the case identifier in the search box. While entering the case identifier, the case(s) that matches the pattern will be displayed in the *List Of Cases* box. To report on reabstraction cases, select the reabstraction option.

To report on a case(s), select the case(s) from the case list and use the **Report On** function to add it to the *Report On Cases* list. One or all of the cases can be selected and also deleted from the Report On Cases list.

Before selecting the <OK> button to produce the report, determine if the report should include all of the variables, just those variables that are filled with values, or variables that have no data entered. Note, the report can take some time to produce depending on how many variables are in the data entry system. Also, if a large number of cases are in the system, selecting all cases may require a large amount of time to produce the reports.

## Data Analysis Variable Report

---

This report lists all of the variables for each table in the selected module in their natural order for which data can be entered. It also lists attributes of the variables that would be necessary to know for performing analysis. These attributes include: field length, data type, format, and definitions for categorical variables. Variables that cannot easily be input into statistical software, i.e., "Memo" fields, are not described but are denoted by "\*\*\*".

## Data Entry Rules Report

---

This report lists the English translation for any of the selected criteria: all variables' rules, one variable's rule(s), or a selected rule type. When all rules are selected, they are listed in the order in which they would be executed during data entry. The report lists the field or screen that causes the rule to execute (trigger field), the field or screen that receives the action (action field or target screen), and the translation. Rules that include a variable that has been deleted from the logic in the module will display the deleted variable as "-".

## Abstractor Summary Report

---

This report measures the provider's and reviewer's timeliness for completing data entry. Selecting this report from the *Select a Report* text box will display the following screen.



The user can report on all of the records or select a set of records based on a specific rule. Check the *Select Records Where* radio button and then click on the <EDIT LOGIC> button. The **Edit Rules** screen will be displayed. By default, the *All* radio button is enabled from the *Select Records to Report on* section. If the user wants to report on only the records based on a specific rule such as a date range edit, he/she can select the *Select Records Where* radio button and then click on the <EDIT LOGIC> button. The **Edit Rules** screen will be displayed. From here, the user can enter in the logic to report on records that are based on a specific rule, i.e., report on record if the admission date is less than or equal to a specified date. See the Edit Rules (Rule Editor) segment in the Design section of the MedQuest Reference manual for a complete discussion of how to add/edit rules.

The user also has the capability to build a list of dates for comparison. Click on the <ADD> button to display the **Select Earlier Date Variable** screen and select a date. After selecting an earlier date variable, the **Select Later Date Variable** screen will be displayed. After selecting a later date variable, the dates will be displayed side by side in the list box. To delete a set of date variables, highlight the set and click on the <DELETE> button. To delete all of the date set variables, click on the <CLEAR> button.

Once the information for this report has been completed, click on the <OK> button to run the report or the <CANCEL> button to cancel the report.

## User Defined Reports

---

MedQuest gives the user the capability to both design and produce the results of a User Defined report.

### Designing Reports

---

This report lets the user design a custom report using the custom grid report similar to the Analyzer tool. After accessing the **Reports** function, click on the <ADD/EDIT USER DEFINED REPORT> button to display the **Edit Reports** screen. From here the user can click on the <ADD> or <EDIT> buttons to display the **Add/Edit Report** screen.

**NOTE:**

The User Defined reports are available for viewing in the MedQuest Data Entry Engine. However, the ability to design a report is only available in the MedQuest Designer.

When adding or editing a report, the Analyzer tool is loaded and the user can work with an indicator that is contained within the reports analysis. Refer to the Analyzer Reference Manual for guidance on how to add/edit an indicator.

## ***Running Reports***

---

When selecting the User Defined Report option, a popup box listing the available reports will be displayed. Select a report and click on the <OK> button to review the results or click on the <CANCEL> button to cancel the report.

The user can report on all of the records or select a set of records based on a specific rule. Check the *Select Records Where* radio button and then click on the <EDIT LOGIC> button. The **Edit Rules** screen will be displayed. By default, the *All* radio button is enabled from the *Select Records to Report on* section. If the user wants to report on only the records based on a specific rule such as a date range edit, he/she can select the *Select Records Where* radio button and then click on the <EDIT LOGIC> button. The **Edit Rules** screen will be displayed. From here, the user can enter the logic to report on records that are based on a specific rule, i.e., report on a record if the admission date is less than or equal to a specified date. See the Edit Rules (Rule Editor) segment in the Design section of the MedQuest Reference Manual for a complete discussion of how to add/edit rules.

## **ISO 11179 Report**

---

This report lists the data entered for all the variables that had ISO attributes assigned to them during the design process. The ISO attributes are reported in a standard format so that the ISO meta data can be passed on to another external system.

---

# GLOSSARY OF TERMS

---

## CORE VARIABLES (MQIS SPECIFIC)

Core variables are variables that are shared by all the MQIS DES Modules. When you design or use a MQIS DES Module, these variables will always appear on the screen in black. They can neither be edited nor deleted.

## DATA ENTRY

A MedQuest function that allows you to collect data for the data entry system designed by the MedQuest Design function by creating the data store file called **XXX.MDB** where "XXX" represents the module acronym (e.g., PNE.MDB is the data store file for the data entry module Pneumonia).

## DATA ENTRY RULE

A rule specified during the design process that is executed during data entry.

## DATA ENTRY SYSTEM

A system designed by MedQuest to collect clinical data for a data analysis project. Each project is called a Module and is represented by a three-character acronym (e.g., PNE is the Data Entry System for the Pneumonia project).

## DESIGN

A MedQuest function that allows you to develop a data dictionary for a data entry system by creating a file called **XXXDICT.MDB** where "XXX" represents the module acronym (e.g., PNEDICT.MDB is the data dictionary file for the data entry module Pneumonia).

## MEDICAL DATA ENTRY DESIGN SYSTEM (MEDQUEST)

An application developed to design a data entry system and to collect data for the system designed.

## MODULE

See **Data Entry System**.

## MODULE VARIABLES

Module variables are variables created by the designer. They are specific to the Module being developed or used.

## SCREENS

A screen is an area beneath the Tab where variables belonging to the same type are laid out (e.g., Tab **Laboratory** represents the Laboratory screen and collects laboratory-related variables). A screen is represented by a Tab. The variables on the screen are displayed by selecting a Tab. A Tab may contain one or more Subtabs.

## **SET OF SCREENS**

The group or groups of screens belonging to a selected DES. If the data entry system to be created or being edited covers a narrowly-defined subject area and contains a reasonable number of variables, only one screen set, the **General Screens**, is necessary.

If the DES being created or being edited covers a discipline or a complex subject that can be logically divided into discrete subject areas, each of these subject areas can be represented by a set of screens (e.g., **Lumbar Disc Disease**, **Head Injury**, and **Brain Tumor** are three sets of screens belonging to the DES called Neurology).

## **SUBSCREENS**

A subscreen is an area beneath the Subtab where variables belonging to the same type are laid out. A subscreen is represented by a Subtab. The variables on the subscreen are displayed by selecting a Subtab.

## **TAB/SUBTAB**

A Tab/Subtab indicates the screen labels (e.g., Tab **History** is the tab that is used for retrieving the **History** screen).

## **VARIABLE TYPES**

Variable types and attributes determine the type of data the system will accept (e.g., a variable will accept date format, etc.) during the data entry. A variable type for each variable is defined by the designer. For example, if variable **Admission Date** accepts only data that are in date form, the variable type must be defined as variable type **Date**.

## **VARIABLES**

A variable is a data entry field that accepts data according to the specifications and rules indicated by the designer during the design.